

IMT Institute for Advanced Studies, Lucca

Lucca, Italy

**Time Series Forecasting Based on Classification
of Dynamic Patterns**

PhD Program in Computer Science and Engineering

XXVII Cycle

By

Rodrigo López Farías

2015

The dissertation of Rodrigo López Farías is approved.

Program Coordinator: Prof. Alberto Bemporad, Institute of Advanced Studies Lucca

Supervisor: Dr. Alberto Bemporad, Institute of Advanced Studies Lucca

Supervisor: Dr. Pantelis Sopasakis, Institute of Advanced Studies Lucca

Tutor: Dr. Alberto Bemporad, Institute of Advanced Studies Lucca

The dissertation of Rodrigo López Farías has been reviewed by:

Dr. Carlos Ocampo Martínez, Universitat Politècnica de Catalunya

Dr. Andrea Emilio Rizzoli, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale

IMT Institute for Advanced Studies, Lucca

2015

Dedicated To My Parents

Contents

List of Figures	x
List of Tables	xiii
Acknowledgements	xiv
Vita and Publications	xv
Abstract	xviii
1 Introduction	1
1.1 Problem Definition	2
1.2 Main Objective	4
1.2.1 Particular Objectives	5
1.3 Justification	5
1.4 Thesis Organisation	7
2 Related Work	8
3 Time Series Forecasting and System Identification	17
3.1 Linear Time Series Analysis	23
3.1.1 Decomposition Methodology in Classical Analysis	28
3.1.2 Stationarity and Sampling	30
3.1.3 Testing for Stationarity	31
3.1.4 Linear Autocorrelations	34
3.2 Box-Jenkins Auto-Regressive Forecasting	36
3.2.1 Auto-Regressive (AR) Models	36

3.2.2	Moving-Average (MA) Models	38
3.2.3	ARIMA(p,d,q)	38
3.2.4	Seasonal ARIMA	39
3.3	Holt-Winters: Exponential Smoothing	40
3.3.1	Single Exponential Smoothing	40
3.3.2	Double Exponential Smoothing	41
3.3.3	Seasonal Holt-Winters	43
3.3.4	Double Seasonal Holt-Winters	44
3.4	k-Nearest Neighbours Forecasting	45
3.4.1	Real Numbers Forecasting	46
3.4.2	Qualitative Forecasting	47
3.4.3	Simple Nonlinear Filter	48
3.5	Radial Basis Function Artificial Neural Networks	49
3.6	Error Measurement Indicators	50
3.7	Classification and Clustering	52
3.7.1	k-Means Clustering	53
3.7.2	Selecting the Best Cluster Partition	53
3.7.3	Feature Extraction	56
4	Multi-Model Forecasting	59
4.1	General Predictor Architecture	61
4.2	Multi-Model Forecasting Using RBF-ANN with an On-line Mode Recognition	62
4.2.1	Discrete Derivative as a Feature Extraction Method	64
4.2.2	On-line Mode Recognition for the Multi-Model Pre- dictor Approach	65
4.3	Multi-Model Predictor Based on Qualitative and Quanti- tative Decomposition	69
4.3.1	Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition of Time Series using SARIMA and kNN	71
4.3.2	Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition of Time Series Using SARIMA and Noise Filter	73

5	Results	79
5.1	Description of the Database	79
5.2	Experiments	84
5.2.1	Validation and Performance Comparison of QMMP Algorithms	85
5.2.2	Validation and Performance of the Multi-Model Fo- recasting Using RBF Artificial Neural Networks with an On-line Mode Recognition	95
5.2.3	Discussion	100
6	Conclusions	102
6.1	General Conclusions	102
6.2	Particular Conclusions	103
6.3	Future Work	105
A	List of Symbols	109
	Bibliography	113

List of Figures

1	Rising, oscillating, and falling patterns classification	10
2	Class 1 descriptor associated with weekend and holiday consumption patterns. (©2014, IEEE).	11
3	Class 2 descriptor associated with weekdays consumption pattern. (©2014, IEEE).	11
4	Phase space representation and clustered in regions of the hourly global horizontal solar radiation ⁰	12
5	Measured hourly global horizontal solar radiation time series of July 1996 ⁰	13
6	Example of an application of ensemble forecasting applied to the prediction of hurricane trajectories.	15
7	In a finite state machine transition states are triggered by events.	22
8	In Markov Model transition states are triggered with certain probability.	22
9	Secular trend	24
10	Seasonal variation	25
11	Noise	25
12	Transient variation	26
13	Additive composition example	27
14	Multiplicative composition example	27
15	Pseudo-additive time series example	28
16	Incomplete pattern	32

17	Four complete patterns	32
18	General structure of the Multi-Model Predictor	62
19	Multi-Model training process	63
20	Time Series processing data	66
21	Multi-model training process	71
22	Prediction sample	74
23	Multi-Model Predictor with the Filter Module	75
24	Raw time series generated by different flowmeters during the year 2012.	81
25	Raw time series generated by different flowmeters during the year 2012	82
26	Linear regression for illustrating the trend of the selected time series	83
27	Linear regression for illustrating the trend of the selected time series	84
28	Autocorrelation plots of different sectors	85
29	Autocorrelation plot of sector p10025	85
30	Organisation of the training-validation and test sets	86
31	Mean silhouette coefficient values for different values of k in k-means and different time series of water demand. . . .	87
32	The two pattern modes of different sectors. The week starts on Sunday.	87
33	Training and testing data organisation	90
34	Error along the testing set of Sector 5	93
35	Prediction sample of 2 days ahead of Sector 5	93
36	Error along the testing set of Sector 11	93
37	Prediction sample of 2 days ahead of Sector 11	93
38	Error along the testing set of Sector 19	93
39	Prediction sample of 2 days ahead of Sector 19	93
40	Error along the testing set of Sector 20	94
41	Prediction sample of 2 days ahead of Sector 20	94
42	Error along the testing set of Sector 78	94
43	Prediction sample of 2 days ahead of Sector 78	94

44	Error along the testing set of Sector 90	94
45	Prediction sample of 2 days ahead of Sector 90	94
46	Error along the testing set of Sector 14	95
47	Error along the testing set of Sector 14	95
48	Mean silhouette coefficient values for different values of k in k -means and different time series of water demand. . . .	96
49	Prediction sample Sector 5	99
50	Prediction sample Sector 11	99
51	Prediction sample Sector 17	99
52	Prediction sample Sector 19	99
53	Prediction sample Sector 20	99
54	Prediction sample Sector 78	99
55	Prediction sample Sector 90	100
56	Mean silhouette value for varying values of k and variable prediction horizon h	100
57	Segments of the generalised decomposition	108

List of Tables

1	Classification of the dynamical systems according to their number of variables	21
2	Number of outliers detected by the modified Thompson <i>tau</i> technique with a significance of $\alpha = 0.01$	83
3	Trend described by the slope and intercept components of linear regression	84
4	Two iteration cross validation for kNN and Calendar mode estimation	88
5	m parameters that optimize the mode prediction with kNN. $\varepsilon = 0.01$ was chosen for all time series	89
6	Parameters for the Noise Filter, kNN Qualitative Forecaster and SARIMA models for each time series found.	91
7	\overline{RMSE}_{24} performance comparison of the different algorithms	91
8	\overline{RMSE}_{24} indicator for each time series produced by QMMP , DSHW , RBFMMP+OR and RBF-ANN	97
9	\overline{RMSE}_h indicator for each method for the hourly time series of the Sector 14.	98

Acknowledgements

I would like to thank my Advisor Dr. Alberto Bemporad, and co-advisor Dr. Pantelis Sopasakis, for their insightful comments and guidance for the thesis dissertation.

Thanks to IMT Institute for Advanced Studies for giving me the great opportunity and resources to finish this PhD.

My sincere gratitude to my external research activity advisor Dr. Vicenç Puig for his continuous support, contribution and close collaboration for this PhD thesis and the facilities provided by the *Universitat Politècnica de Catalunya*.

Thanks to Dr. Juan J. Flores from *Universidad Michoacana de San Nicolás de Hidalgo* for his scientific support and contribution to this thesis and the invitation to collaborate during the final period of my PhD. Thanks also to the Electrical Engineering Faculty from the same University for providing me the facilities required to finish this important project.

Thanks to my friend and colleague Dr. Héctor Rodríguez Rangel for his participation, support and contribution.

Thanks to Dr. Carlos Ocampo M. from *Universitat Politècnica de Catalunya* and Dr. Andrea E. Rizzoli from *Istituto Dalle Molle di Studi sull'Intelligenza Artificiale* for the useful and careful revision of this thesis.

Also thanks to Elsevier and IEEE for granting permission for including some material from their sources contained in *Journal Energy Conversion and Conference proceeding Management, Control and Automation (MED)*.

Thanks to my parents Lucrecia and Victor for their unconditional support and patience. Now we know that great decisions are not usually easy to take.

Vita

- July 8, 1984** Born, México D.F., México
- 2008** B. Eng. Computational Systems
Final mark: 86/100
Instituto Tecnológico de Morelia, México
- 2010** MSc. in Electrical Engineering
Final mark: 93/100
Posgrado de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo México

Publications

1. R. Lopez, V. Puig, H. Rodriguez, "An implementation of a multi-model predictor based on the qualitative and quantitative decomposition of the time-series", in *International work-conference on Time Series 1*. Granada, Spain, 2015.
2. R. Lopez, V. Puig, "A Multiple-Model Predictor Approach Based on an On-Line Mode Recognition with Application to Water Demand Forecasting" in *International work-conference on Time Series 1*. Granada, Spain, 2015.
3. R. Lopez, J. Flores and V. Puig, "Multi-Model Forecasting Based in a Qualitative and Quantitative Decomposition with Nonlinear Noise Filter and an Application To Water Demand" in *2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* . Ixtapa, Mexico, 2015.
4. J. Flores, J. Ortiz Bejar, J. Rafael Cedeño, C. Lara-Alvarez and R. Lopez, "FNN a Fuzzy Version of the Nearest Neighbor Time Series Forecasting Technique" in *2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* . Ixtapa, Mexico, 2015.

Presentations

1. R. López, “An Hybrid Algorithm based on Particle Swarm Optimization with Niches and Quasi Newton for Searching Fixed Points in Non-Linear Dynamical Systems,” at *10th State Congress of Science, Technology and Innovation by the State council of Science and Technology. 10mo Congreso Estatal de Ciencia, Tecnología e Innovación (CECTI)*, Morelia, Mexico, 2015.

Abstract

This thesis addresses the problem of designing short-term forecasting models for water demand time series presenting nonlinear behaviour difficult to be fitted with single linear models. These behaviours can be identified and classified to build specialised models for performing local predictions given an estimated operational regime. Each behavior class is seen as a forecasting operation mode that activates a forecasting model. For this purpose we developed a general modular framework with three different implementations: An implementation of a Multi-Model predictor that works with Machine Learning regressors, clustering algorithms, classification, and function approximations with the objective of producing accurate forecasts for short horizons. The second and third implementations are hybrid algorithms that use qualitative and quantitative information from time series. The quantitative component contains the aggregated magnitude of each period of time and the qualitative component contains the patterns associated with modes. For the qualitative component we used a low order Seasonal ARIMA model and for the qualitative component a k -Nearest Neighbours that predicts the next pattern used to distribute the aggregated magnitude given by the Seasonal ARIMA. The third implementation is based on the same architecture, assuming the existence of an accurate activity calendar with a sequence of working and rest days, related to the forecast patterns. This scheme is extended with a nonlinear filter module for the prediction of pattern mismatches.

Chapter 1

Introduction

In areas like natural sciences, economics or engineering it is necessary, for specific purposes, to monitor or observe the dynamics of certain phenomena that is related to the field of study. The weather dynamics, industrial process, the fluctuation of the stock market are just some examples where the understanding of the dynamics is relevant. For example, the study of the environment dynamics in natural sciences is useful for the implementation of policies to preserve ecosystems, optimise the use of natural resources and improve the quality of life by modelling the dynamics of the urban sprawl.

The prediction of the stock market in economics is vital for making better decisions about the actions that can be taken by the investors, and the study of model identification with the objective of constructing a model that behaves similarly to the real process for control. There are examples where the observation and modelling of the dynamics of the system is relevant, and a subset of them are related directly to the study of time series.

Time series are presented explicitly or implicitly in every day life. A time series is defined as a sequence of data measurements chronologically ordered with certain frequency. This data might come from different sources related to the studied discipline; these data may come from human activity, wind dynamics, and mathematics, among others. These

disciplines follow different ends but they share in common the problem: the modelling of the dynamical system that fits better with the observed data able to produce or simulate such information. One of the most active research for these purposes is system modelling for prediction.

The study of the analysis of time series was born with the need to understand the dynamics of the data fluctuation generated by an unknown system. Dynamics are seen as changes of values along time of certain variable of study. These fluctuations might represent different kinds of data, depending on the application field.

The observed data is generated by a known or unknown model. When the model is unknown, a general dynamical model is constructed from previous analysis of the data.

A general classification of the models used for forecasting can be done according to the linear nature of their structure. The classification according to this criterion is:

- **Linear models:** Explain the relation between the variables by means of linear correlation.
- **Nonlinear models:** The relation between the variables are not explained by means of linear correlations. The modelling deals with a nonlinear structure, present in piece-wise linear and nonlinear models. The characteristics of these models are:
 - **Piece-wise linear models:** Is a set of linear models that are activated when certain conditions are satisfied.
 - **Piece-wise nonlinear models:** Is a set of nonlinear models that are activated when certain conditions are satisfied.

1.1 Problem Definition

Time series forecasting is performed by a regression function, which is a model that receives a sequence of observations and returns a scalar or a vector of real numbers. The regression function that predicts the value

in the next instant of time $t + 1$, given a sequence of values in \mathbf{Y}'_t is expressed by Equation 1.1.

$$\hat{\mathbf{Y}}_{t+1} = \mathbf{F}(\mathbf{A}, \mathbf{Y}'_t) \quad (1.1)$$

where \mathbf{F} is the regression function, $\mathbf{A} = \{a_1, \dots, a_k\}$ are the parameters of the model, \mathbf{Y}'_t is the input vector with m number of elements $\mathbf{Y}'_t = \{Y_{t-m}, \dots, Y_t\}$ and $\hat{\mathbf{Y}}_{t+1}$ is the prediction given by the regressor \mathbf{F} returning h steps ahead $\hat{\mathbf{Y}}_{t+1} = \{\hat{Y}_{t+1}, \dots, \hat{Y}_{t+h}\}$. The general objective function for fitting time series with a regressor model is given by Equation 1.2.

$$\min_{\{\mathbf{A}\}} \sum_{t=m}^{n-h} \|\hat{\mathbf{Y}}_{t+1} - \mathbf{Y}_{t+1}\|^2 \quad (1.2)$$

where the regression function minimises the squared errors between the output of the regressor \mathbf{F} and the original data $\mathbf{Y}_{t+1} = \{Y_{t+1}, \dots, Y_{t+h}\}$.

Time series are generated by dynamical systems from different sources, such as energy sources as wind (JQWS15), energy prices (XPX11), (US14), human water demand, and water precipitation (Wat03) often difficult to model and forecast with precision. According to the stationarity and sampling theory usually systems present an intermittent change of behavior (HA13) that implies a mismatch with the regression model. This problem motivates the use of alternative regression methods and combinations of them, depending on the nature of the system.

The problem to solve in this thesis is the design of a general framework that incorporates multiple regression models that are selected to be activated according to predefined rules. For this purpose, the data is analysed and clustered in classes according to their common characteristics to fit local models. The proposed model to study is given by the piecewise Equation 1.3

$$\mathbf{F}(\mathbf{A}, \mathbf{Y}'_t) = \begin{cases} f_1(a_1, \mathbf{Y}_t) & \text{if } mode = 1 \\ f_2(a_2, \mathbf{Y}'_t) & \text{if } mode = 2 \\ \vdots & \\ f_k(a_k, \mathbf{Y}'_t) & \text{if } mode = k \end{cases} \quad (1.3)$$

where $\mathbf{F}(\mathbf{A}, \mathbf{Y}'_t)$ is the multi-model that contains k independent local models f_1, \dots, f_k . \mathbf{A} is the set of the selector parameters for the models and it is defined as $\mathbf{A} = \{a_1, a_2, \dots, a_k\}$ and \mathbf{Y}'_t is a vector with recent measurements at time t . The objective of study is to build a multi-model that fits complex dynamics of time series. The objective function for the multi-model is described by Equation 1.4.

$$\arg \min_{\{\mathbf{A}\}} \sum_{t=m}^{n-h} \|\mathbf{F}(\mathbf{A}, \mathbf{Y}'_t) - \mathbf{Y}_{t+1}\| \quad (1.4)$$

The selection of the modes of global modelling is a design problem for the activation of different modes according to knowledge collected from the observed data and a priori information. The developing of this global modelling is related to the construction and exploitation of probabilistic or deterministic rules that should be explored for finding a suitable model of this kind that estimates accurately the next operation mode.

1.2 Main Objective

The main objective is to find high performance drinking water demand prediction models that provide accurate predictions in the short term. The availability of an accurate and detailed prediction is a very important part for making accurate decisions regarding the operation, control and management of drinking water networks. An accurate model allows minimising operational costs and wastewater without sacrificing quality of service, delivering drinking water to the population. The problem is addressed studying the identification and classification of different dynamic patterns found in drinking water demand time series for the design of local predictors that are integrated later in a global model.

1.2.1 Particular Objectives

- To explore the integration of machine learning, data mining, and statistical models such as neural networks, k-means clustering, and Box-Jenkins models in Multi-Model predictors (**MMP**) and compare their performance with classical forecasting methods such as exponential smoothing and traditional neural networks for data regression.
- To explore and validate with standard metrics the clustering of time series to identify different behaviours and its decomposition to simplify and improve the accuracy of the forecasting models.
- The exploration and design of global methodologies for the detection and activation of forecasting operation modes.
- Test and validate in the short term the Multi-Model Predictor architecture with drinking water demand time series.

1.3 Justification

Time series analysis is an important discipline useful in the optimisation of the exploitation of natural resources and renewable energy. The performance of model forecasters impacts directly the operation costs since an estimate of future information can be used to take optimal decisions in the management of drinking water. For example, in the case of the drinking water delivery, where special attention is required in the maximisation of the water availability and minimisation of the operational costs for bringing safe drinking water to the population, it is important to have certainty about the future requirement of this resource in different terms (short and long term) to optimise its use.

The Primary Health Care in Alma-Ata declared in 1978 the safe water as the most important resource for human health (Org93). The extraction, treatment, storage, and distribution of drinking water is a costly and complex task usually bringing it from faraway places using pipe networks connected to the urban population to distribute the consumption

of this element (AS77). In the operational cost of the water production are implied chemicals, legal canons, electricity costs. The transportation of the drinking water also contributes to the electricity cost since the water pumping stations require energy to operate, such as the Barcelona drinking water delivery (BRRP11).

Optimising the management of the water network supply also avoids the unnecessary expansion of the water network infrastructure and new supplies (S^{+98}). It also reduces withdrawals from limited freshwater supplies, reducing at the same time the negative effect that produces the exploitation of this resources on the natural environment.

To manage the water supply network efficiently several strategies have been developed. One of them is Model Predictive Control (MPC) (PRP⁺13), an optimisation-based control strategy applicable to a wide range of industrial applications (YBH⁺12; OL94; OJ06). MPC provides suitable techniques to compute optimal control strategies ahead in real time for all the control elements of a water supply system. The accuracy of MPC depends on the water distribution model and the accuracy of the short term forecasting water demand. MPC solves the control problem each time step finding the best input control sequence several steps ahead, applying just the first action of the sequence. Since the MPC uses the prediction as reference for the optimal control, inaccurate predictions increase statistically the operation cost. According to the study of Hippert et al. in (HPS01) an increase of 1% in the error would imply to £10 million increase of operational cost.

Usually MPC is extended with a feedback mechanism that deals with the system disturbances. This extension consists of solving the best input sequence for a certain forecast horizon $t + 1, \dots, t + h$, and apply just the solution of the next step $t + 1$. It is desired to apply this process each time t in an environment with disturbances. The longer the horizon, the better control performance is achieved. MPC applied to drinking water networks has as main objective to reduce operational costs, related to production, transportation and the maximisation of the quality of service, delivering the water properly to the population. For this reason it is important to make accurate water consumption predictions which will

be used by MPC.

1.4 Thesis Organisation

The thesis is organised as follows: Chapter 2 addresses related work to the Multi-Model forecasting framework. Chapter 3 introduces time series forecasting and system identification, considering the linear and nonlinear approaches for the analysis also including an introduction of classification and feature extraction important for data treatment that can be implemented straightforwardly in the proposed framework. Chapter 4 addresses the proposed Multi-Model Predictor architecture and three proposed implementations. Chapter 5 presents the results of the different proposed forecasting methodologies. Finally Chapter 6 presents conclusions and future work suggested by the author.

Chapter 2

Related Work

In the early successful stage, during the 70's decade, important discoveries appeared in time series modelling and forecasting with the first applications in econometrics. George E.P. Box and G.M. Jenkins (BJR94) used the *divide and conquer* strategy decomposing and characterising the basic components of series trying to explain in a certain way the characteristics of the general dynamics of time series, such as trend, seasonal, cyclic, and random components. All these components were integrated for the first time in the Autoregressive Integrated Moving Average methodology creating the Box-Jenkins or ARIMA methodology.

The first reference regarding the study of the combination of forecasts produced by different models is found in 1976. It asserts that although the single Box-Jenkins forecasting is better than other methodologies of that time, like exponential smoothing, a simple average forecast from a set of models can be more accurate under some circumstances. This implies the suggestion of using a combination of several models instead of one. This discovery by Castaño et. al (CM00) stimulated the study of the linear combination of forecasts models to improve the prediction performance.

The study affirms that after proving under the assumption of having different unbiased forecasts, the optimum linear combination of forecasts produces another unbiased forecast. In order to optimise the weights

of the linear combination it is necessary to have as much evidence as possible for constructing the forecast.

This idea was accepted gradually with the development of expert systems, which reinforced the idea of combining different forecasters belonging to different information sources. Castaño (CM00) also statistically proved that the forecast produced by a combination of models is always better than the use of a single model.

Nowadays, the development of a new generation of forecasting models and strategies is strongly related to multidisciplinary novel research mainly from mathematics, computing, and statistics. Regarding the time series literature, there is a strong effort on finding the best way to decompose time series in several but simpler time series to fit better simpler models that improve the forecasting performance. This is not an easy task since in real cases arise several challenges, one is about the unavailability of a full model that describes the dynamic fluctuation of the data. Often the time series information is insufficient, noisy, or corrupted. For this reason the data should be analysed and processed to be fixed. In other situations the data is so large and complex that it is computationally infeasible to optimise parameters of statistical models or training the machine learning models.

Fortunately, despite all the problems that may occur, thanks to the growing of computational resources and the development of machine learning and pattern recognition algorithms, it is possible to analyse time series with a higher complexity in their dynamics (e.g. Nonlinear dynamical systems and hybrid dynamical systems). A good review about the representation, indexing, similarity measure, segmentation and visualisation of time series analysis from the data mining point of view is found in (Tc11). Also the book of Multiple Model Approaches to Modelling and Control (SJ97) a comprehensive survey about modelling using multiple models is presented. A collection of practical examples and approaches are discussed where the single modelling approach is not enough for systems with complex and hybrid motion.

Although the multi-modelling approach was born with the analysis of partially known systems (grey box modelling), the same ideas can be

adopted for time series which there is no knowledge about the system or mathematical model that produces its dynamics. Real cases are presented regularly in the water demand, solar radiation, wind speed and stock market fluctuations.

As a historical reference, one of the earliest works related to water time series forecasting, presents a multi-model application to water demand forecasting shown in the work of Shvartser in (SSF93). This work proposes a methodology based on pattern recognition and time series analysis; the daily consumption cycle is divided in three segments; rising, oscillating, and falling. These segments are modelled separately and are seen as dynamical states. The sequence of the activation of each pattern associated with one state is modelled with a Markov process capturing the transition probabilities between states. Each segment of the time series with a specific pattern is associated with each state which is represented with a low order ARIMA model.

Figure 1 is an example of the decomposition of the time series in segments. Segments number 1, 2, and 3 are classified as rising, oscillating, and falling respectively where the limits of the segments are defined by the dense dotted lines. Each segment may occur during the day transaction indicated by the pointy line. The pattern is observed throughout the information of two days.

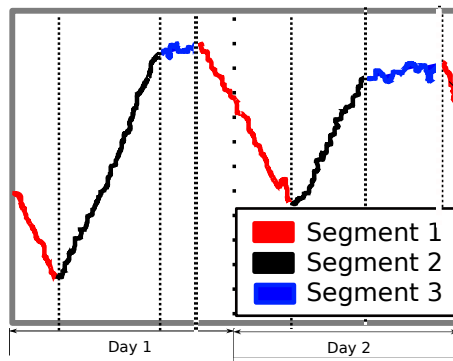


Figure 1: Rising, oscillating, and falling patterns classification

S. Alvisi and M. Franchini (AFM07) developed a short-term, pattern-based model for water-demand forecasting. The model captures the periodic patterns presented at annual, weekly and daily levels.

The model structure is based on the observed patterns at different abstraction levels of the water demand time series. The forecasting structure is hierarchically organised in two levels: The high level module that captures the low frequency patterns, like the seasonal and weekly patterns of the time series observed in Figure 1 from (AFM07). The low level model describes and predicts the daily consumption (Figure 3 presented in (AFM07)). In order to get the hourly forecasts over the next 24 hours period, a short term forecasting mechanism based on the combination of both models is implemented.

The work of J. Quevedo and V. Puig in (QSPB14) addresses a similar approach, where a Seasonal ARIMA (SARIMA) is used for predicting the daily water demand consumption combined with a descriptor class that distributes the amount of the predicted water consumption along one day. Mainly two validated descriptors are used. These descriptor classes are shown in Figure 2 and Figure 3, describing working and resting days patterns. These descriptors are validated using the LAMDA clustering method found in SALSA software package (KAS⁺03; Kem04). The patterns given by the descriptors are a priori assigned to each calendar day according to the human activity calendar.

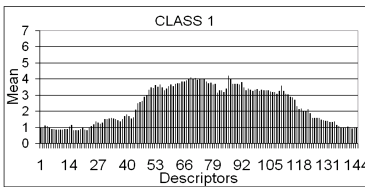


Figure 2: Class 1 descriptor associated with weekend and holiday consumption patterns. (©2014, IEEE).

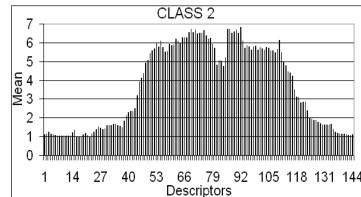


Figure 3: Class 2 descriptor associated with weekdays consumption pattern. (©2014, IEEE).

Benmouiza developed (BC13) a methodology based on the combination of clustering methods and artificial neural networks to predict in the short term the solar horizontal radiation. The predictor model is a composition of several independent local models. Each model is trained with clustered data of one class containing similar dynamical patterns. The architecture has a Global Nonlinear Autoregressive Neural Network (NAR) used for predicting the local model to forecast. Once the local model is selected, a local NAR model associated with each cluster is used to forecast the hourly radiation. Although the application of the work aims to forecast the solar radiation, it might be possible to implement it for water demand forecast, since the dynamics of the water demand may obey also to global and local structure in its dynamics where the local dynamic patterns can be identified and clustered.

Figure 4 shows how the data in phase space is classified using a clustering algorithm finding three kind (or regions) of solar radiations. Each region represents low, medium and high solar radiation levels. A global NAR (Neural Auto Regressive) predicts the next region. Once the next region is estimated the local model forecaster related to the region is used to predict the radiation in an hourly basis. The dynamics of the hourly basis radiation is shown in Figure 5.

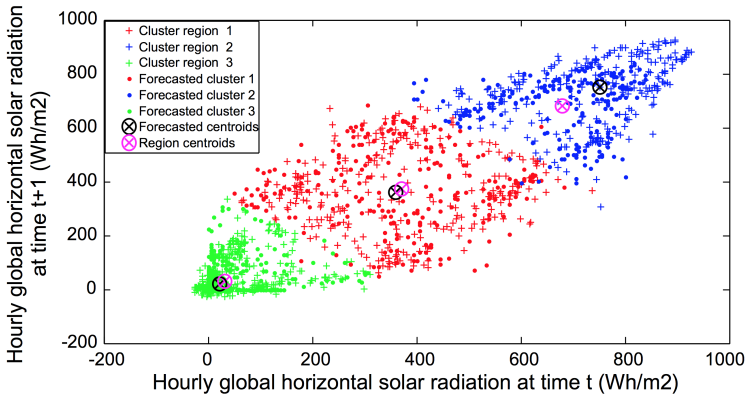


Figure 4: Phase space representation and clustered in regions of the hourly global horizontal solar radiation 0 .

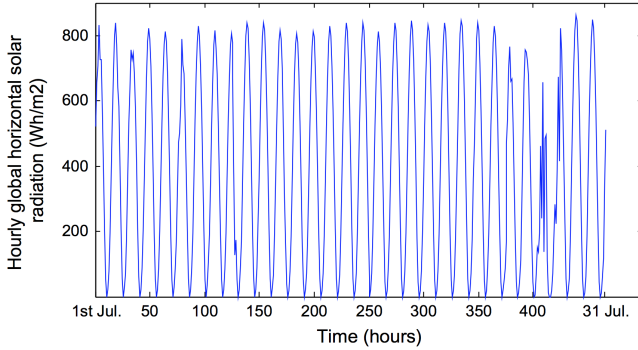


Figure 5: Measured hourly global horizontal solar radiation time series of July 1996⁰.

M. Bakker et al. (BVVS13) propose a fully adaptive forecast scheme using a static calendar to compute in real time model weight coefficients (named day factors), and demand patterns used by the model. The model assumes the existence of four kinds of different water demand patterns (reported in (ZAWL02)) of which those associated with holidays, weekdays, and holidays variations are known in advance but not the variants of season water demand patterns which should be detected on-line.

Another interesting approach based on the combination of multiple models, are the consensus and ensemble methods. The consensus models are tools to create structured prediction maps which consider a limited set of future forecasts based on expert information. This set of forecast are provided by the human knowledge (McN87). On the other hand, an ensemble forecast is a collection of two or more forecasts performed at the same time. These methods focus on generating scenarios that describe probabilistically the predicted states of a dynamical system (LP08). Ensemble forecasting is considered a Monte Carlo analysis method where

⁰Energy Conversion and Management, Vol 75, K. Benmouiza, A. Cheknane, Reprinted from Forecasting hourly global solar radiation using hybrid k-means and nonlinear autoregressive neural network models, Pages No 561-569, Copyright (2013), with permission from Elsevier

multiple numerical predictions are produced from generating different possible initial conditions given a past sequence and current set of observations. Applications of these models are found in medicine (KOB13), health (FKCB84), economics, meteorology and water management. In economics, this approach is so relevant at the point that exists a specialised firm named Consensus EconomicsTM group (Dat16) that collects the state-of-the-art forecasters with their predictions for a big number of variables (more than 1000) from 85 industrialised countries in Eastern Europe, Asia Pacific and Latin America. The group has a significant community of researchers that confirm better accuracy of Consensus ForecastsTM than most of the individual forecasters (Bat00; BWWA01; Jon14; NR11).

In meteorology, consensus forecasts are implemented to predict weather and meteorological phenomena. As an example, Figure 6, a picture taken from the National Weather Service (Ser15), describes a category 4 Hurricane, Debby, that appeared in 1994. Each colored trajectory line, is a forecast simulation considering a randomised scenario. The set of forecast trajectories are used to produce a nominal prediction. This forecast is the most likely trajectory that might be taken as official to predict the phenomenon dynamics.

Considering the survey of novel application in water management by Donkor et al. (DMSR14), the general concern of forecast methodologies is to fit the time series reducing the difference between the real value and the estimated forecast. Even though model fitting by minimising this gap is crucial to select a good forecasting model, the uncertainty prediction is also an important component for the forecast that should be considered in order to define the prediction bounds to give confidence to the prediction. With regard to water management, Tiwari et al. (TA15) propose a bootstrap method to learn a wavelet based machine learning considering also the minimisation of the prediction bounds. Other articles studying the uncertainty of stochastic models are found in (HK15), (CCK⁺08) and (AF14).

Based on the multi-modelling approach and the references collected regarding the design of multi-model forecasting, this thesis proposes the

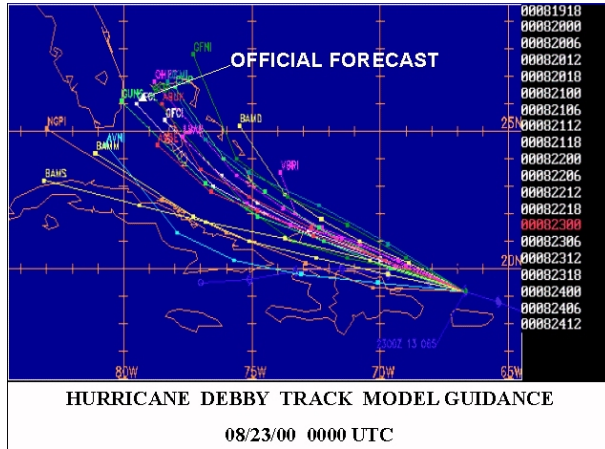


Figure 6: Example of an application of ensemble forecasting applied to the prediction of hurricane trajectories.

next contributions to multi-model forecasting.

- The design of a module based framework that is able to exploit the empirical information using machine learning algorithms rather than the seasonal structure.
- Algorithms based on the exploitation of the historical information are useful as alternative to the existing modelling global methods.
- An implementation based on machine learning algorithms such as neural networks and clustering that identifies on-line different kinds of behavior patterns.
- Two implementations based on the qualitative and quantitative decomposition of the time series. Where the predicted quantitative information is distributed along of an unitary pattern that describes a kind of activity (e.g, working or resting days).
 - The first variation does not assume any a priory human information about the water consumption modes and proposes

the use of a qualitative k-Nearest Neighbour (kNN) for mode prediction.

- The second implementation assumes the existence of an activity calendar used as predictor, but with the contribution of extending the method using a simple nonlinear filter to detect the qualitative pattern mismatches to readjust the pattern forecast along time.

This Chapter presented a brief historical introduction where George E.P. Box and G.M. Jenkins explored the decomposition of the data to analyse and understand time series dynamics to construct a stochastic model capable to predict time series with certain accuracy. After some research Castaño et al. proved that a combination of unbiased forecasts gives regularly a better unbiased forecast. Nowadays computational power gives the possibility to analyse and construct complex dynamics presented in time series. We addressed recent research in multi-modelling applied to time series from real sources as water demand, solar radiation, stock market fluctuations and meteorology. Next Chapter 3 addresses the common forecasting methods and clustering algorithms oriented to system identification for our proposal.

Chapter 3

Time Series Forecasting and System Identification

A time series is defined as a sequence of chronologically ordered observations recorded at regular time intervals. The observations are sequential data that might represent qualitative or quantitative information depending on the source and application field. When the data are quantitative, the measurements that compose the time series express magnitudes or scalar information. An univariate time series notation used in the literature is defined as follows in Equation 3.1:

$$\mathbf{Y}_{full} = \{Y_1, Y_2, \dots, Y_t, \dots, Y_n\} \quad (3.1)$$

where \mathbf{Y}_{full} is the full time series, t is the index that indicates the time when the value was taken, n is the length of the time series. Usually time series \mathbf{Y}_{full} , deals with scalar numbers that might be integers, reals, but sometimes they store qualitative information, where each element takes on a symbol from a defined set of objects. $Y_t \in \mathbf{A}$, where \mathbf{A} is a set of symbols included in the alphabet.

These measurements are taken via observations and then recorded somehow; for example manually or using computers that interact with electronic devices that record the information in data bases. The use of high speed computers and big storage allows to implement powerful

statistics and machine learning algorithms.

Time series forecasting is strongly related to system identification and modelling. System identification integrates statistical and mathematical tools to estimate and exploit the available information, and also studies the optimal design of experiments to generate informative data to fit dynamical models. This is achieved by removing redundant or erroneous data keeping useful and descriptive information to model the object of study. This data treatment is useful for reducing the training or the parameter optimisation time, and the model simplification.

The selection of information is important to control the detail level of the model according to the power and capacity of the implementation device: e.g., for small devices with limited memory and power processing a simple version of the model generated from the data should consider the most important characteristics of the object of study (as models implemented in microcontrollers for embedded control).

There are three levels of modelling abstraction in system identification. According to the scope of the and knowledge availability of the study object, the modelling levels can be classified as white, grey, and black-box modelling.

In white-box modelling, the dynamical processes are modelled using differential equations to describe the motion of the system. Under this approach there is detailed and enough information to build an accurate model describing the nominal motion of the system of interest. This kind of modelling is useful for solving the implementation problems of control, simulation and synthesis of the control law, especially in the space state approach. Typically, system identification is done by a human expert following laws and rules, for example the physical law of motion or chemical reaction laws. Although white-box is desirable in many cases due to the parsimony of the models, this approach has limitations when the systems are more complex or when it is not possible to have the precise parameter values for the system. In these cases the white-box methodology is not enough.

The next abstraction level of modelling is grey-box modelling. It addresses those problems where the dynamical system is partially known

and the model is completed using empirical information. A typical case of grey-box modelling is when the general dynamic of the system is modelled but there are parameters that still need to be tuned. For example, the parabolic shot modelled with newton gravitational laws or hydrologic model behaviours such as Nash-Sutcliffe and coefficient of determination model efficiency (KBB05) (used to assess the predictive power of hydrological models). In this case, these parameters should be estimated using statistical methods with the available observations so far.

Black-box modelling is the highest level of modelling abstraction of the system identification approaches where there is little, no information, or unclear insight about the model behind that generates the sequence of observed data. This kind of modelling uses general purpose models tuned or trained using just the observed data. Examples of this kind of models are Auto Regressive (AR) models, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) for discrete time modelling of continuous dynamical systems, or deterministic state machine for piecewise models or Markov Chains Models for piecewise stochastic models.

Once a model is obtained, it can be applied to simulation, control or forecasting of dynamical systems among others. The concept of black-box modelling abstraction is linked to the time series analysis that is the main focus of this thesis.

The main purpose of the time series analysis, is the design of dynamical models based on the empirical observations of different phenomena. For the study of these observations, time series analysis provides the useful theory for the construction of methodologies and algorithms that are provided as tools for understanding the information for the correct modelling of the behaviour of the observed data.

During the analysis of time series two general approaches are taken into account depending on the linearity of the data, like the linear and nonlinear approaches. Linear methods are useful when the interpretation of the observed data is regular presenting a dominant frequency and sequential information that can be measured with linear correlations. It is assumed that the systems are governed and explained with linear alge-

bra theory. Using this approach the linear equations are limited to model systems that present a decaying, growing or damped periodically oscillating behaviour. The remaining irregularities are assumed to be random external inputs or disturbances to the system that can be described statistically by the normal distribution. Some examples of systems studied under this assumption are shown in Table 1 taken from the book *Nonlinear Dynamical Systems* in (Str94) of Strogatz and Steven H., where the dynamics of the linear systems is determined and classified by the number of variables or differential equations. The basic systems belonging to this classification, present a simple growth, decay, or equilibrium dynamics when the dynamical systems contain one variable. When oscillations are present in the systems, they can be modelled with linear systems of two variables, for example, a simple mass-spring system. With more than three variables, applications are found in engineering e.g., coupled systems modelled with several linear systems. For more complex systems like coupled oscillators require a greater number of equations or variables to model them.

On the other hand, nonlinear methods address a more general family of dynamical models. This approach must be considered when the time series presents irregularities from the point of view of linear dynamical systems theory. The main drawback of linear theory is the impossibility to distinguish the random noise from the nonlinear structure of the data, therefore, it cannot model appropriately this kind of behaviour since the nonlinearity produces false residuals that might be confused with the noise where in reality it can be still modelled somehow. The systems in this category are natural processes. The simplest nonlinear models that contain just one variable are used to study fixed points, bifurcations, over damped systems or the equilibrium in ecosystems. For two variables natural oscillations, like limit cycles, biological oscillations, and nonlinear electronics are studied. For three or more variables, systems that might present strange attractors, chemical kinetics, iterated maps or fractals are studied. For many variables the nonlinear optics, non equilibrium statistical mechanics, heart cell synchronisation, biological neural networks response, complete ecosystems and economics behavior are studied. For

the continuum domain, nonlinear waves, plasma, earthquakes, general relativity, quantum field theory, reaction diffusion, biological and chemical waves among others systems are studied.

Table 1: Classification of the dynamical systems according to their number of variables

n	$n = 1$	$n = 2$	$n > 3$
General approach	Growth, decay, or equilibrium	Oscillations	Applications in engineering
Linear	Exponential growth RC circuit		Civil Engineering Electrical Engineering
Nonlinear	Fixed points Bifurcations Over damped systems Relaxational dynamics Logistic equation for single species	Pendulum Limit cycles Biological oscillators (Heart cells, neurons) Nonlinear electronics	Strange attractors 3-Body problem Chemical kinetics Iterated maps Fractals Forced nonlinear oscillators Practical uses of chaos

n	$n \gg 1$	Continuum
General approach	Collective phenomena	Waves and patterns
Linear	Coupled harmonic oscillators Equilibrium in statistical mechanics	Elasticity Wave equations Electromagnetism Acoustics
Nonlinear	Lasers, non linear optics Non-equilibrium statistical mechanics Nonlinear solid state physics Heart cell synchronisation Neural networks Immune systems Ecosystems Economics	Nonlinear waves Plasma Earthquakes General relativity Quantum field theory Reaction diffusion Biological and chemical waves Fibrillation Epilepsy Turbulent fluids Life

All the mentioned systems are able to generate a time series since it is possible to measure the states of certain variables along time. For this reason the study of time series is also strongly related to the study of nonlinear dynamical systems when they come from natural or artificial systems.

The present thesis is related to dynamical systems difficult to model analytically belonging to nonlinear systems family. The models are obtained using different technics with the final objective of producing accurate forecasts.

A special case of nonlinear systems are the piecewise linear or nonlinear dynamical systems, where the system is modelled by a set of local models that are activated by certain rules. Each local model is associated with one operation mode. The activation of these local models depends on the model of the finite state machine in the deterministic case (Bla08), or by a stochastic state machine that activates the different local mod-

els by probabilistic rules such as those performed by Markov Models Chains. Figure 7 shows an example of a finite state machine model that captures the dynamics of behavior changes. The submodels or states are activated when an event occurs. In contrast, the Markov Model (GS10) in Figure 8 models the change of states by producing probabilities events to jump from one state to another.

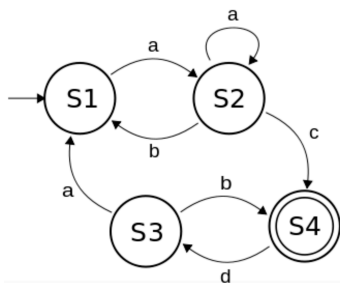


Figure 7: In a finite state machine transition states are triggered by events.

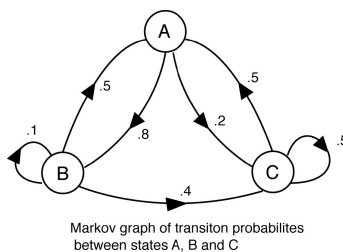


Figure 8: In Markov Model transition states are triggered with certain probability.

The next subsection addresses the basics of linear time series analysis introducing its basic components, a general methodology to extract them, and some notes about stationarity and sampling that remarks the importance of selecting the correct data to the study and modelling of time series.

3.1 Linear Time Series Analysis

Linear time series analysis studies time series that can be modelled by analysing and identifying the secular trend, seasonality, and noise components. A popular and simple way to analyse the dynamics is by using Auto-Regressive time series models. These models are a standard in modern stationary time series data analysis (MJK08). The advantage of these models is that they are seen as a combination of components of larger models that leads to generalised forms.

Although it has limitations for nonlinear time series, the concepts and structure of linear models provide a background for the analysis of nonlinear models. Univariate time series also can be classified according to their domain in frequency based methods and Time domain based methods.

Regarding time domain based methods the time series are studied from the stochastic process point of view. A stochastic process is a sequence of random variables Y_t taking any value from $[-\infty, \infty]$ where t is interpreted as the time in the discrete domain. Given a sequence of values each of the Y_t variables have their own function that captures their distribution within of their corresponding moment. Each pair of these variables will have their corresponding joint distribution and the marginal distribution functions.

These time series can be decomposed to be analysed by its three components: the trend (long term direction), the seasonal (systematic, calendar related movements) and the irregular (unsystematic, short term fluctuations) components.

In order to proceed with the description of some of the most popular methods used in the literature as Box-Jenkins and Holt-Winters, basic definitions are defined next about the structure of the linear time series.

- i Secular trend is the persistent component of the observed phenomena; it describes the long-term trend. The appearance of this component is exemplified in Figure 9 where the red line shows the long-term trend. Although for simplification a linear fitting is used, it is also common to observe a long-term nonlinear trend (e.g., exponen-

tial or geometric growth). Some examples of secular trend cases can be global warming, inflation in economy, increase of energy and consumption and decrement of the availability of the water resources along the time.

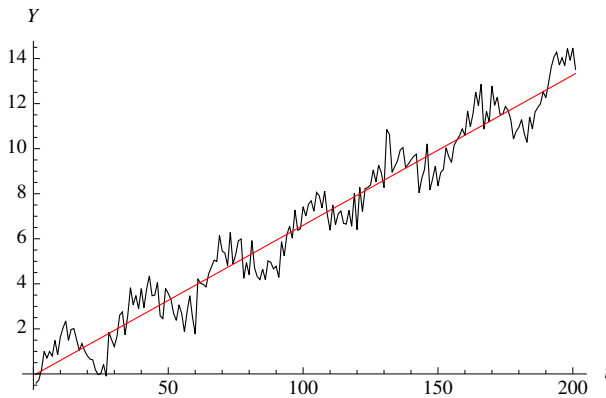


Figure 9: Secular trend

- ii Seasonality variation describes the short-term periodic movement. An example of how this component looks is presented in Figure 10. Some examples where this component is present are daily variation of the temperature, daily sea level and water demand in a short term.
- iii White noise is modelled by a normal distribution function that captures the random variation of variables. This component also has the property of having zero mean, constant and independent variance for different values along time. An example of what the noise component looks is shown in the Figure 11. White noise can be presented as external disturbances, error measurements due to technical limitations and interference.
- iv Transient variation captures the accidental dynamic presented regularly as isolated perturbations or aperiodic fluctuations that affect the regular behaviour over the time. An example of the appearance

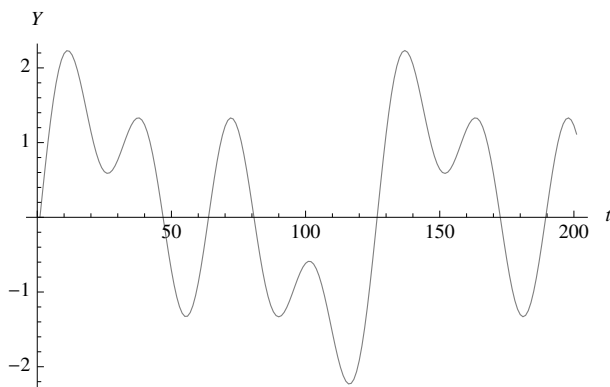


Figure 10: Seasonal variation

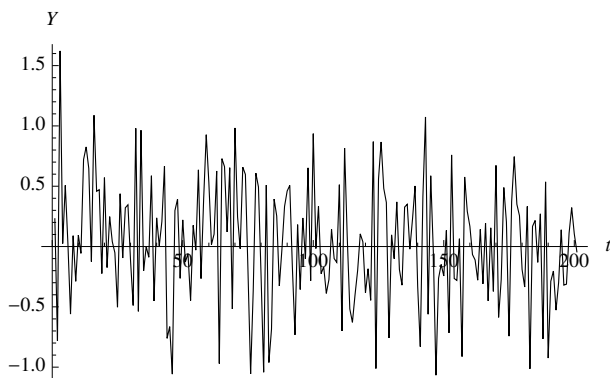


Figure 11: Noise

of this component is shown in Figure 12, where aperiodic oscillations are present. It is possible to confuse this component with noise. When it appears, a nonlinear model might be used for describing the dynamics of this component.

In the time series classical analysis literature, decomposition is used to describe separately the trend and seasonal factors. Linear modelling can be classified according to the way of combining these components

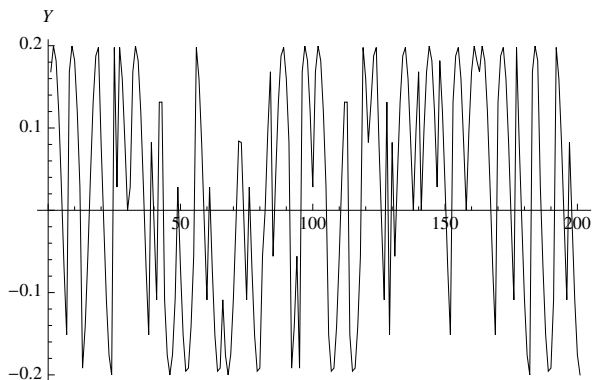


Figure 12: Transient variation

in additive, multiplicative and pseudo additive compositions (HA13). Another kind of decompositions focuses on describing long-run cycles, like the weekends and holiday effects (QSPB14). The classification of time series according to their decomposition is:

- i Additive models consider the sum of the trend, seasonal, and random variation. It is used when the seasonal variation is relatively constant over time. It is expressed by Equation 3.2

$$Y_t = T + S + R \quad (3.2)$$

where the times series is generated by adding T , S , and R , the trend, seasonal, and random components respectively. An example of the general structure of this model is shown in Figure 13.

In this figure the red line is the trend approximation of the time series, and the black line is the real data with seasonal components and noise.

- ii Multiplicative models are composed by the multiplication of the different components like trend, seasonal and random components. This kind of model is used when an increasing seasonal variation is observed along the time. It is expressed by Equation 3.3.

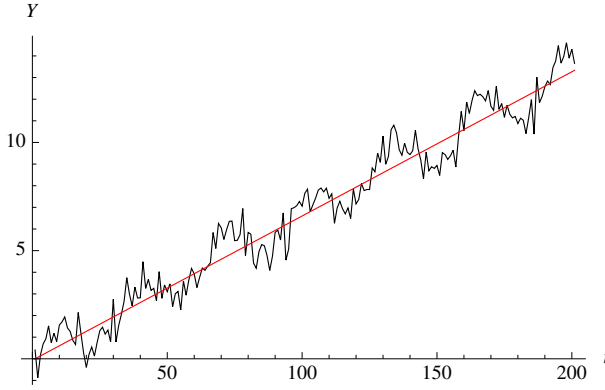


Figure 13: Additive composition example

$$Y_t = T \times S \times R \quad (3.3)$$

where the components R , S and T are the same described previously, with the difference that the time series Y_t is the result of the product of such components. An example of the structure using this kind of decomposition is shown in the Figure 14.

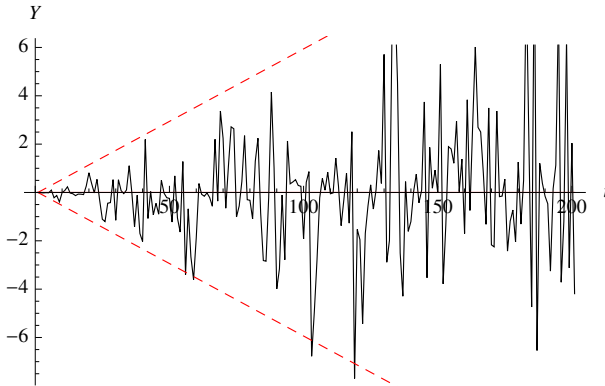


Figure 14: Multiplicative composition example

The red dotted lines of the figure shows the multiplicative trend of

variance of the time series Y_t .

- iii The pseudo-additive decomposition (Mixed) presents a more complex dynamic, adding and multiplying the different components. It combines features of both the additive and the multiplicative models. The general structure of this kind of model is expressed by Equation 3.4:

$$Y_t = TC_t \times (S + I) - 1 \quad (3.4)$$

where the addition of the seasonal effects, and irregular fluctuations $(S + I) - 1$ produces a trend in the mean, and the trend cycle component TC_t produces a multiplicative effect that increases the variance along time. Figure 15 shows an example of how a time series of this type looks like. The low and high red dotted line bound the time series according to an increasing variance. The continuous red line is the mean increasing along the time.

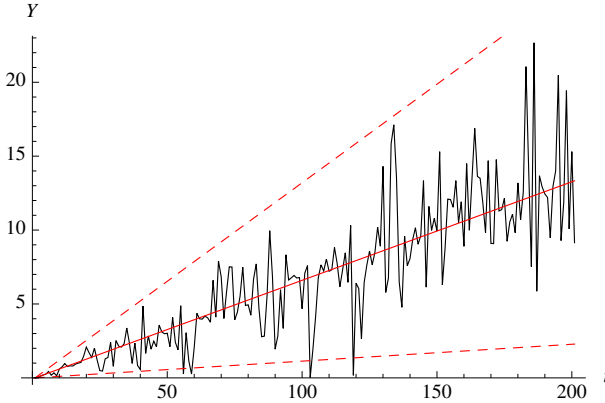


Figure 15: Pseudo-additive time series example

3.1.1 Decomposition Methodology in Classical Analysis

In time series classical analysis, it is important the study of different components present in time series separately. Although time series analysis

may be considered a sort of art, it requires statistical knowledge, theory and good practices for generating a valid and correct model. Box and Jenkins proposed a methodology to understand systematically how a time series behaves following a *divide and conquer* strategy (BJR94). Decomposing the data in simpler time series is crucial to identify and estimate the different components involved in the time series. An important assumption to consider is the generation of the data by a linear dynamical system. Considering the different components the general steps that should be taken into account, according to Box and Jenkins are:

- i Observation: This step is the most important because with visual analysis is possible to have the intuition and information about general aspects of the data useful for selecting the proper general model to use.
- ii Trend estimation: To detect the estimation in the long term in time series can be used mainly two different kind of methods belonging to different approaches:
 - Filtering: It is a function capable of neglecting the short term dynamics and keep the general dynamics. An example of this kind of algorithms is the smoothing effect of a moving average.
 - Estimating a regression equation: A (linear or nonlinear) function might be used to describe the trend using least squares optimisation (Sch13). (Like the solid red line in Figures 9, 13 and 15).
- iii De-trending: When the trend is already known, for the additive composition the trend is removed subtracting it from the original time series. For the multiplicative composition the trend is removed dividing the time series by the trend.
- iv Detection of seasonal factors: If the seasonality is known, the simplest method for estimating these effects is to average the de-trended values for a specific season using the Seasonal Sub-series Plot, otherwise a correlation plot helps finding the seasonality (Cle93) .

- v Determine the random (irregular) component: The random component is analysed last. After obtaining the forecasting model, the variance of the noise by means of gaussian random distribution is analysed.

series. Matlab does this (and estimates the trend with a straight line in the iteration.

3.1.2 Stationarity and Sampling

In order to study a phenomena it is important to reproduce it many times under the same conditions and be sure that the measurements taken effectively correspond to the same scientific study, object, or process. The concept of reproducibility in time series is strongly related to invariability and availability meanings.

1. Invariability of parameters during the analysis of the system: These parameters must not depend on time and they change once they are noticed or desired to produce intentionally an output.
2. Availability of the data addresses the problem of the incorrect modelling due the lack of information which impedes to detect and consider the non-stationarity of the process. Non-stationarity is difficult to detect and model since it is also difficult to know if the acquired information is enough and reliable providing a good description of the general dynamic of the data.

When the phenomena is studied a finite number of times and behaves differently we can also say that it represents a non-stationarity process, and presents an intermittency effect. The intermittence is observed only if we can replicate the process enough times. Conversely, the analyst will not have enough data to produce a reliable model. Therefore, the length of the series must provide enough information to describe and take into account intermittency.

3.1.3 Testing for Stationarity

The stationarity test is performed over stochastic processes. The simplest non-periodic stochastic process, is composed by a set random variables $\{Y_t\}_{t \in \mathcal{O}}$, where $Y_t \in \mathbb{R}$, t is the time index included in the set \mathcal{O} , with infinite values in the interval $(0, \infty]$ in a continuous-time. For discrete-time process the random variable takes a finite number of \mathbb{N} values $\mathcal{O} = (0, \mathbb{N}]$, restricting the length of the discrete time series. A stationarity process is present when the probabilities related to the random variables are constant over time. This is expressed in Equation 3.5.

$$Pr(Y_{t+\tau}) = Pr(Y_t) \quad (3.5)$$

where Pr is the probability of observing certain value, and $\{Y_t, Y_{t+\tau}, \dots, Y_{t+p_s\tau}\}$ are the observations belonging to the same probability distribution $Pr(Y_t)$.

Stationarity is confirmed when there is no violation of the basic properties of the stochastic system like variance, mean, transition probabilities, and correlations. Considering the difficulty for detecting stationarity for periodic time series, it is recommended to get considerably longer data length than the period length for modelling to capture as much cyclic samples as possible. As soon as we get large data, we will have more chances to determine and distinguish the global trend, patterns and intermittence effects. This means that the sample data size must be greater than the period, τ .

In other words we need a large data set of size n , such that $n \gg p_s\tau$, where τ is the size of the period and p_s is the number of periods that guarantee p characteristic samples, e.g., Figure 16 gives the impression of non-stationarity only if the data we have is limited, therefore it is not possible to determine periodicity. On the other hand, if we have more data, like in Figure 17 a pattern is revealed. Hence local stationarity is detected. Figure 17 shows complete cycles with a relative stationarity (since we do not know what will happen in the future).

Assuming we have sufficient information, we can take two approaches for testing stationarity: the parametric and nonparametric app-

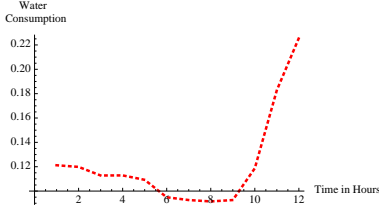


Figure 16: Incomplete pattern

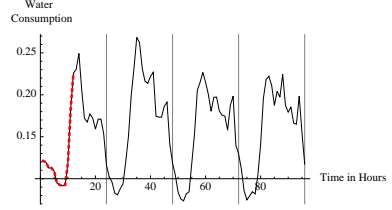


Figure 17: Four complete patterns

roach. The parametric is usually used by researchers considering statistical assumption about the distribution of the data in the time field domain, such as economists and statesmen. Such assumptions can be tested according to the 1st or 2nd order stationarity criteria.

- 1st order stationarity is related to strict stationarity criteria. It assumes a time-independent joint statistical distributions and variations of the time series, therefore, the mean and variance at any moment must be the same. Strict stationarity satisfies Equation

$$[Pr(Y_{t_1+\tau'}), \dots, Pr(Y_{t_{n_s}+\tau'})] = [Pr(Y_{t_1}), \dots, Pr(Y_{t_{n_s}})]$$

where the probability of the shifted vectors satisfies the equality independently of the chosen value $n_s, t_1, \dots, t_{n_s} \in \mathcal{O}$, or lag τ' (LRS13).

In real time series this definition is very tight and time series presenting some irregularities almost never satisfy this condition.

- 2nd order stationarity is about relaxed stationarity condition. It is more flexible and considers a stationary process only if the mean (first moment) is constant and covariance (second moment) is finite and depends just on the time difference $\tau' = t - t'$ along time series.

The nonparametric approach does not require any assumption of the data, therefore it is more general. One of the most popular stationarity

test is the Runs Test. The test defines a run as “succession of one or more identical symbols or patterns”, which are followed and preceded by a different symbol or no symbol at all (Gib83). The idea behind is similarly to a series (or runs) generated by identical flips of a coin, where the symbol O and I represents heads and tails, respectively. An example of such run is the sequence $\mathbf{C} = \{OIIIIIOOOIOO\}$. If the coin is balanced, the sequence will present stationarity approximating the same number of O and I as long as more observations are available, otherwise a trend might be detected.

The stationarity of data can be determined by using the algorithm of the runs test proposed by Bendat and Piersol in (BP86) as follows:

1. Divide into intervals T of length τ .

$$Z'_T = \{Y_t\}_{t=\tau(T-1)+1}^{T\tau} \quad (3.6)$$

2. Compute a mean value for each interval.

$$\bar{Z}'_T = \frac{1}{\tau} \sum_{t=1}^{\tau} (Z'_T) \quad (3.7)$$

3. Collect the occurrences of values above (I) and below (O) the median value \tilde{Z}' of the series using Equation 3.8.

$$C_T = \begin{cases} I & \text{if } \bar{Z}'_T > \tilde{Z}' \\ O & \text{otherwise} \end{cases} \quad (3.8)$$

4. Count the number of runs with I and O following Equation 3.9

$$\begin{aligned} C^O &= |\{i, \forall i \in \{1, 2, \dots, T-1, T\}, |C_i = O\}| \\ C^I &= |\{i, \forall i \in \{1, 2, \dots, T-1, T\}, |C_i = I\}| \end{aligned} \quad (3.9)$$

5. Compute the probability of having a run $Pr(\mathbf{C}^O)$ and $Pr(\mathbf{C}^I)$ and if $Pr(\mathbf{C}^O) \approx Pr(\mathbf{C}^I)$, the the stationarity test is passed.

3.1.4 Linear Autocorrelations

We have seen so far how the general characteristics of the time series are analysed discarding dynamic details at local level. In order to analyse what is going on with the observed dynamic fluctuations at local level of the time series, a relation analysis between variables is performed.

The relation analysis is an important part of the study of linear time series, it focuses on the existence of a linear temporal relation between the the variable of interest and another one is named autocorrelation.

There are two kinds of autocorrelation metrics used as tools for measuring the existence and intensity of such correlation among the data, the simple and partial autocorrelation.

- The Simple Autocorrelation Function (ACF), measures the lineal relation between the observation Y_t from a time series and the delayed value $Y_{t-\tau}$.
- The Partial Autocorrelation (PACF), is the estimation of the simple correlation but removing the effect produced by the autocorrelations for delays shorter than τ .

The degree of relation for the simple and partial autocorrelation is measured using the autocorrelation coefficient ρ_τ . The coefficient takes values in the interval $[-1, 1]$. This coefficient provides information about the existence and the type of correlation. A coefficient with value of $+1$ means a strong positive relation between two observations separated by k units of time. The positive sign means that both variables respond similarly. A value of -1 expresses a strong inverse relation. When a value of 0 is obtained, the two variables do not exhibit any relation. Equation 3.10 computes the autocorrelation function.

$$\rho_\tau = \text{corr}_{\tau_\tau}(Y_t, Y_{t+\tau}) = \frac{\gamma_\tau}{\gamma_0} \quad (3.10)$$

where corr is the autocorrelation function at lag τ , ρ_τ is the coefficient of the simple autocorrelation for a delay of τ , and γ is the covariance function defined in Equation 3.11

$$\gamma_\tau = \text{cov}(Y_t, Y_{t-\tau}) = E[(Y_t - \bar{Y})(Y_{t-\tau} - \bar{Y})] \quad (3.11)$$

where cov is the covariance function, \bar{Y} is the mean of the time series, $Y_{t-\tau}$ is the observation at lag τ , E is the expectation and γ_0 is equal to 1 by definition following Equation 3.12

$$\gamma_0 = \text{var}(Y_t) = \rho_0 = 1 \quad (3.12)$$

where var is the variance. This property is found in (Fra85).

The Partial Autocorrelation Function (PACF) of τ order is similar to the ACF that measures the correlation between two variables Y_t and $Y_{t-\tau}$, but with the difference of discarding the influence of the dependency of the intermediate lags between both of them. The partial autocorrelation gives information about the AR order. The recursive formula proposed by Levinson and Durbin (Fra85) is described in Equation 3.13

$$\begin{aligned} \hat{\pi}_{11} &= \rho_1 \\ \hat{\pi}_{\tau,\tau} &= \frac{\rho_\tau - \sum_{j=1}^{\tau-1} \hat{\pi}_{\tau-1,j} * \rho_{\tau-j}}{1 - \sum_{j=1}^{\tau-1} \hat{\pi}_{\tau-1,j} \rho_{\tau-j}} \end{aligned} \quad (3.13)$$

where $\hat{\pi}_{\tau,j} = \hat{\pi}_{\tau-1,j} - \hat{\pi}_{\tau,\tau} \hat{\pi}_{\tau-1,\tau-j}$ is the covariance at different lags $j = 1, 2, \dots, \tau - 1$.

Equation ACF 3.10 and PACF 3.13 are used to produce plots named autocorrelograms. Each autocorrelogram is composed plotting the coefficients ρ_τ and $\hat{\pi}_{\tau,\tau}$ respectively varying τ at different lags. Each autocorrelogram provides information about the Moving-Average (MA) and Auto-Regressive (AR) polynomial order and structure (BJR94). The description of the Box-Jenkins ARIMA methodology is addressed in Chapter 3.2.

The Ljung-Box test is an auxiliary test born in 1970 (BP70) and implemented to quantify the model fitting (LB78) measuring the independency of the values of time series. This test gives information about the possible existence of linear relations in the data or if the data sequence is merely random noise. The Ljung-Box test is given by Equation 3.14

$$LB_m = n(n+2) \sum_{t=\tau}^m \frac{\hat{\pi}_{t,t}}{n-\tau} \quad (3.14)$$

where n is the sample size, m is the lag length and $\hat{\pi}_{t,t}$ is the autocorrelation of the sample at lag t .

The LB_m value is evaluated using the χ^2 test. If LB_m exceeds the critic value of the χ^2 to the significance level selected (usually at 95%). The test assumes the null correlation hypothesis H_0 among the values until the opposite is demonstrated. If the alternative is demonstrated, then H_0 would be rejected and the existence of significant correlation hypothesis H_1 is taken. This is expressed in Equation 3.15.

$$\begin{cases} H_0 : \text{No Rejection of null hypothesis} & \text{if } LB_m < \chi^2_{(m)} \\ H_1 : \text{Rejection of null hypothesis} & \text{otherwise} \end{cases} \quad (3.15)$$

3.2 Box-Jenkins Auto-Regressive Forecasting

A very useful framework for time series analysis is the Box-Jenkins Auto-Regressive method. In 1970 Box and Jenkins proposed this framework to identify, estimate, and diagnoses dynamic models of time series, where the variable time is important. The objective of Box-Jenkins is to provide an algorithmic way to construct forecasting models (AM10).

3.2.1 Auto-Regressive (AR) Models

The first set of models to capture the dynamic of linear time series is the Auto-Regressive model. Auto-regressive models explain an endogenous variable of a period t by the previous observations adding an error term (under previous conditions, Y_t can be expressed as a linear combination of its past values plus an error term). The notation of the Auto-Regressive models are abbreviated as AR models, a p -order model is written as $AR(p)$. Where p is the number of observations in the past that are taking into account by the model. The basic Auto-Regressive model is expressed as $AR(1)$ and is represented by Equation 3.16.

$$\hat{Y}_t = \Phi_0 + \Phi_1 Y_{t-1} + \epsilon_t \quad (3.16)$$

where \hat{Y}_t is the forecast value, ϵ_t is the error term with the property of having a zero mean, constant variance, and null covariances among errors associated with different observations. The general AR(p) model is expressed by Equation 3.17

$$\hat{Y}_t = \sum_{i=1}^p \Phi_i Y_{t-i} + \Phi_0 + \epsilon_t \quad (3.17)$$

where the Φ_i components are the Auto-Regressive coefficients, p is the order of the model and ϵ is assumed to be a random variable with normal distribution with zero mean. This can be simplified using the lag polynomial operator composed by lag operators that are applied to move back the index t , p times backward presented in Equation 3.18

$$\begin{aligned} L^1 Y_t &= Y_{t-1} \\ L^2 Y_t &= Y_{t-2} \\ &\vdots \\ L^p Y_t &= Y_{t-p} \end{aligned} \quad (3.18)$$

Therefore, taking into account the lag operator definition, Equation 3.19 defines the lag polynomial operator $\Phi_p(L)$,

$$\Phi_p(L) = 1 - \Phi_1(L) - \Phi_2(L)^2, \dots, \Phi_p(L)^p \quad (3.19)$$

where $\Phi_0 = 1$. This allows to have a compact representation of the AR process as shown in Equation 3.20.

$$(1 - \Phi_1 L - \Phi_2 L^2 - \dots - \Phi_p L^p) \hat{Y}_t = \Phi(L) \hat{Y}_t = \epsilon_t \quad (3.20)$$

More details about the lag notation and manipulation is found in (HA13).

3.2.2 Moving-Average (MA) Models

The Moving-Average (MA) model explains the value of a variable at time instant t , as a weigh function of the previous independent error terms. The model is expressed as $MA(q)$ where q is the number of error terms. The basic Moving-Average model is expressed as $MA(1)$ and is written as Equation 3.21:

$$Y_t = \mu + \epsilon_t + \Theta_1 \epsilon_{t-1} \quad (3.21)$$

where μ is the mean and Θ is the coefficient of the model and ϵ_{t-1} is the previous prediction error. Equation 3.21 can be generalised as shown in Equation 3.22.

$$Y_t = \sum_{i=1}^q \Theta_{(t-1)\epsilon_{(t+1-i)}} + \mu \quad (3.22)$$

and the lag polinomial notation is used as well to express any $MA(q)$ model as shown in Equation 3.23.

$$\Theta_q(L) = 1 - \Theta_1(L)^1 - \Theta_2(L)^2, \dots, \Theta_q(L)^q \quad (3.23)$$

Letting $\Theta_0 = 1$, the MA process can be written in a compact form similarly to AR process, as in Equation 3.24

$$\hat{Y}_t = \Theta(L)\epsilon_t \quad (3.24)$$

3.2.3 ARIMA(p,d,q)

The ARMA model is a case of the ARIMA model that includes the Auto-Regressive and Moving-Average components. It is written as $ARMA(p, q)$ where p is the order of the Auto-Regressive component and q of the moving average component (Equation 3.25)

$$\hat{Y}_t = \Phi_0 + \epsilon_t + \sum_{i=1}^p (\Phi_i Y_{t-i}) + \sum_{i=1}^q \Theta_i \epsilon_{t-i} \quad (3.25)$$

the lag polinomial notation is used to represent the model

$$\Phi(L)(\hat{Y}_t - \mu) = \Theta(L)\epsilon_t \quad (3.26)$$

The nonseasonal ARIMA is written as ARIMA(p, d, q) where d is the degree of differencing associated with the component of integration $(1 - L)^d Y_t$. This component discards any non stationary trend from the time series and is present in Equation 3.27.

$$\left(1 - \sum_{i=1}^p \Phi_i(L)^i\right) (1 - L)^d Y_t = \left(1 + \sum_{i=1}^q \Theta_i(L)^i\right) \quad (3.27)$$

3.2.4 Seasonal ARIMA

The general version of the Box-Jenkins models is the Seasonal ARIMA (SARIMA) model. This model also considers the modelling of the seasonal components of the time series. This model uses two polynomials expressed by ARIMA (p, d, q) (P, D, Q) $_{\bar{s}}$. Where the notation component (p, d, q) is the nonseasonal part of the model and (P, D, Q) $_{\bar{s}}$ is the seasonal component where m is the number of measurements per season or cycle. The order of the different polynomials are expressed by p, d, q, P, D , and Q . In order to illustrate how the seasonal component extends the non-seasonal model, we analyse a SARIMA model $(1, 1, 1)(1, 1, 1)_{\bar{s}}$ presented in Equation 3.28:

$$(1 - \Phi_1(L))(1 - \phi(L)^{\bar{s}})(1 - L)(1 - (L)^{\bar{s}})Y_t = (1 + \Theta_1(L))(1 + \theta_1(L)^{\bar{s}})\epsilon_t \quad (3.28)$$

where the seasonal AR, MA, and the difference components consider a seasonal lag at time \bar{s} in the component $(L)^{\bar{s}}$ that multiplies their respective nonseasonal components.

The Box-Jenkins model optimisation parameters is a well studied topic, therefore it is possible to find statistical optimisation methodologies and their implementation in different software packages. Regarding the optimisation methods, one of the most popular methodologies are the minimisation of the exact maximum likelihood (HM88), and least

squares fitting methods. The software packages that provide these functionalities are found in *statsmodels* (*Python*), Econometrics Toolbox (*MATLAB*), *ARIMAProcess* Function (*MATHEMATICA*), and the *arma* command (*STATA*), among others.

3.3 Holt-Winters: Exponential Smoothing

An alternative to the Box-Jenkins forecasting is the exponential smoothing method. The basic framework of this method was proposed in the late 50's by C.C. Holt in 1957 and his student Peter Winters (Goo10). Basically, the forecast produced by this method is the result of weighing the average of the past observations with weights decaying exponentially as the observations get older. The advantages related to this framework is to generate forecasts easily with very low storage requirements, ease of automation, reliable and adaptable to new changes in trends and seasonal patterns. These characteristics make these methods suitable to be applied widely in the industry due to their simplicity (Kal04).

3.3.1 Single Exponential Smoothing

The basic Holt-Winters comes from a family of exponential smoothing methods based on the continuously updated forecast giving more relevance to the recent experience. Exponential smoothing is named in this way because the weights decrease exponentially as the observations gets older. The single exponential smoothing is used for short term forecasting. It assumes that the dynamics of the data fluctuate around a reasonably stable mean. That is, the assumption of lack of trend or seasonal components of the time series under analysis. Given the information available in time t , the exponential smoothing is given by Equation 3.29.

$$\hat{Y}_{t+1} = \sum_{i=0}^{\infty} w_i Y_{t-i} \quad (3.29)$$

where $W = \{w_0, w_1, \dots, w_{\infty}\}$ are the weights given to the past values of the series with the property of summing 1. The recent values have a

higher weight than the old ones. The weights are set using Equation 3.30

$$w_i = \alpha(1 - \alpha)^i \quad (3.30)$$

where $i = \{1, 2, \dots, \infty\}$ and α is a constant real value $0 < \alpha < 1$ that regulates the relevance of the previous values to produce the forecast. Each successive observation in the series is a new smoothed value. This value is computed as the weighted average of the current observation and the previous smoothed observations. Substituting Equation 3.30 in 3.29 are obtained the forecasting Equations 3.31-3.33.

$$\hat{Y}_{t+1} = \alpha Y_t + \alpha(1 - \alpha)Y_{t-1} + \alpha(1 - \alpha)^2 Y_{t-2} + \dots + \alpha(1 - \alpha)^m Y_{t-m} \quad (3.31)$$

where m is the number of considered lags. Factoring Equation 3.31 we obtain:

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)(\alpha Y_{t-1} + \alpha(1 - \alpha)Y_{t-2} + \dots + \alpha(1 - \alpha)^{m-1} Y_{t-m}) \quad (3.32)$$

which is expressed iteratively in Equation 3.33

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t \quad (3.33)$$

where \hat{Y}_t is the previous estimate. An alternative notation of the smoothing forecasting is given by Equation 3.34

$$\hat{Y}_{t+1} = \alpha \epsilon_t + \hat{Y}_t \quad (3.34)$$

where $\epsilon_t = Y_t - \hat{Y}_t$. It works as the correction for Equation 3.33.

3.3.2 Double Exponential Smoothing

Double exponential smoothing is an extension of the single exponential smoothing that considers the trend in time series. The exponential smoothing component that captures the linear trend is described by the linear Equation 3.35

$$\hat{Y}_t = \ell_{t-1} + b_{t-1} \quad (3.35)$$

where the terms ℓ_t and b_t are the level and slope (trend) at time t . This model is used to predict stable long-term time series. The h -step ahead prediction, also named Holt's method, is performed by Equation 3.36

$$\hat{Y}_{t+h} = \ell_{t-1} + hb_{t-1} \quad (3.36)$$

where h is positive and defines the prediction at time h . A deeper analysis of this component is found in (Tho80). The level and trend components are smoothed according to Equation 3.37-3.38

$$\ell_{t-1} = Y_{t-1} + (1 - \alpha)^2 \epsilon_{t-1} \quad (3.37)$$

$$b_{t-1} = b_{t-2} + \alpha^2 \epsilon_{t-1} \quad (3.38)$$

where ϵ_{t-1} is the previous forecasting error defined by Equation 3.39

$$\epsilon_{t-1} = \hat{Y}_{t-1} - Y_{t-1} \quad (3.39)$$

The Holt-Winters linear trend estimates the trend using a weighting function which considers more relevant the most recent observations. This estimation uses a local trend equation where the linear trend ℓ and slope b components of the forecasting Equation 3.36 are updated each period with Equation 3.40 and 3.41.

$$\ell_t = \alpha Y_t + (1 - \alpha_1)(\ell_{t-1} + b_{t-1}) \quad (3.40)$$

$$b_t = \alpha_2(\ell_t - \ell_{t-1}) + (1 - \alpha_2)b_{t-1} \quad (3.41)$$

Similar to single exponential smoothing, the slope and level components also can be expressed as a function of their errors using the correction form presented in Equation 3.42 - 3.43.

$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha_1 \epsilon \quad (3.42)$$

$$b_t = b_{t-1} + \alpha_2 \epsilon \quad (3.43)$$

The problem of the initialisation of parameters ℓ_1 and b_1 has been addressed before (Kal04). Different methods have been proposed for the initialisation of these values. For b_1 , it is possible to use the alternatives given by Equation 3.44 - 3.46, for ℓ_1 it can be initialised to Y_1 .

$$b_1 = Y_2 \quad (3.44)$$

$$b_1 = [(Y_2 - Y_1) + (Y_3 - Y_2) + (Y_4 - Y_3)]/3 \quad (3.45)$$

$$b_1 = (Y_n - Y_1)/(n - 1) \quad (3.46)$$

The three initialisation options approximate a trend in different ways; the first one is by taking just the second element of the time series, the second is computing the average of the three first differences of the time series, and as third option is considered the slope produced by the division of the subtraction of the first element Y_1 of the last element Y_n .

3.3.3 Seasonal Holt-Winters

The Holt-Winters extension for seasonal components in time series is composed basically by three equations: the level equation ℓ_t , the trend equation b_t and the seasonal component equation s_t^1 . With smoothing parameters α_1 , α_2 , and α_3 , and the seasonal parameter τ that expresses the period of the seasonality. These components are described in Equation 3.47-3.50

$$\hat{Y}_{t+h} = \ell_t + hb_t + s_{t-\tau+h_\tau^+}^1 \quad (3.47)$$

$$\ell_t = \alpha_1(Y_t - s_{t-\tau}^1) + (1 - \alpha_1)(\ell_{t-1} + b_{t-1}) \quad (3.48)$$

$$b_t = \alpha_2(\ell_t - \ell_{t-1}) + (1 - \alpha_2)b_{t-1} \quad (3.49)$$

$$s_t^1 = \alpha_3(Y_t - \ell_{t-1} - b_{t-1}) + (1 - \alpha_3)s_{t-\tau}^1 \quad (3.50)$$

where $h_\tau^+ = [(h - 1) \bmod \tau] + 1$.

The level equation computes an average of the weights between the seasonal measurement ($Y_t - s_{t-\tau}^1$) and the nonseasonal prediction ($\ell_{t-1} +$

b_{t-1}) in time t . The trend equation, is similar to Equation 3.40. The seasonal equation is an average of the weights between the current seasonal time, $(Y_t - \ell_{t-1} - b_{t-1})$, and the same seasonal time of the previous year. The corresponding error correction alternative is given by Equations 3.51 - 3.53.

$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha_1 \epsilon_t \quad (3.51)$$

$$b_t = b_{t-1} + \alpha_1 \alpha_2 \epsilon_t \quad (3.52)$$

$$s_t^1 = s_{t-m}^1 + \alpha_3 \epsilon_t \quad (3.53)$$

where $\epsilon_t = (\ell_{t-1} + b_{t-1} + s_{t-\tau}^1) = Y_t - \hat{Y}_{t|(t-1)}$ are the one-step training for the forecast errors.

3.3.4 Double Seasonal Holt-Winters

The double seasonal Holt-Winters (DSHW) is an extension of the previous method adding an extra seasonal equation (s^2) (TdMM06), that captures a second seasonal period. It is also added other smoothing and seasonal parameter ω and τ_2 respectively. The new parameters have information about a bigger periodicity multiple of τ_1 . The additive DSHW is described in Equation 3.54 - 3.58.

$$\hat{Y}_{t+h} = (\ell_t + hb_t) + s_{t-\tau_1+h}^1 + s_{t-\tau_2+h}^2 \quad (3.54)$$

$$\begin{aligned} \ell_t &= \alpha_1(Y_t - (s_{t-\tau_1}^1 + s_{t-\tau_2}^2)) + \\ &\quad (1 - \alpha)(\ell_{t-1} + b_{t-1}) \end{aligned} \quad (3.55)$$

$$b_t = \alpha_2(\ell_t - \ell_{t-1}) + (1 - \alpha_2)b_{t-1} \quad (3.56)$$

$$s_t^1 = \alpha_3(Y_t - (\ell_t + s_{t-\tau_2}^2)) + (1 - \alpha_3) + s_{t-\tau_1}^1 \quad (3.57)$$

$$s_t^2 = \omega(Y_t - (\ell_t + s_t^1)) + (1 - \omega) + s_{t-\tau_2}^2 \quad (3.58)$$

Where Y_{t+h} is the h -th step ahead forecast from forecast origin t . The parameters are optimised by minimizing the sum of squared errors of one step-ahead residuals. In order to use the multiplicative version the

Single and Double Seasonal Holt-Winters, they are obtained by the product of the seasonal components (SS11; SBM07).

3.4 k-Nearest Neighbours Forecasting

The k-Nearest Neighbours (kNN) is an nonparametric learning algorithm used in classification and forecasting. It takes decisions based on the experience contained in the training set (in the best case a subset of them). kNN assumes that the data is in a feature metric space. The considered data might be scalars, multidimensional vectors, labels or characters, etc. kNN forecasting is used to solve the problem of estimating the next elements Y_{t+1}, \dots, Y_{t+h} given a set of examples of the time series. kNN is considered a lazy learning algorithm which given a sample of past information, finds in the history similar scenarios to imitate their dynamics and use them to construct a short term forecast. kNN only needs three parameters: the sampling rate τ , the embedding dimension m , and the geometric size of the neighbourhood defined in ε (where the objects inside the radius ε are considered part of the neighbourhood). kNN forecasting requires a times series defined in Equation 3.59

$$\mathbf{Y}_{full} = \{Y_1, \dots, Y_t, \dots, Y_n\} \quad (3.59)$$

where Y_1 is the first element of the time series, Y_n is the latest value at time n and t is the t -th measurement. For this purpose, the time series is organised in subsequences named delay vectors of the form.

$$\mathbf{D}_t = \{Y_{t-(m+1)\tau}, \dots, Y_t\}$$

where m is the size of the delay vector \mathbf{D}_t , and τ is the sampling frequency over the time series where $t = \{(m-1)\tau + 1, (m-1)\tau + 2, \dots, n-1-h\}$. To get the nearest neighbours of \mathbf{D}_t , the parameter ε is defined to get the delayed neighbour vectors v_1, \dots, v_k respect with \mathbf{D}_t satisfying Equation 3.60

$$dist(\mathbf{D}_{v_k}, \mathbf{D}_t) \in \varepsilon \quad (3.60)$$

where *dist* can be any distance. For magnitudes is common to use the Euclidean distance, but for comparing sequences of qualitative information, the Hamming distance is an adequate metric. For real numbers *dist* can be any norm. The more used norms are infity and euclidean distance.

3.4.1 Real Numbers Forecasting

The forecasting for real numbers where the vector to forecast is a real number vector $\mathbf{D}_t \in \mathbb{R}^m$ is performed by using Equation 3.61 with the vectors that satisfy the constraint expressed in Equation 3.60. The forecasting value is the result of the weighted sum of the *futures* $Y_{t+h\Delta n}$ of the selected vectors v_1, \dots, v_κ and it is defined in Equation 3.61

$$\hat{Y}_{t+h\Delta n} = \alpha_j \sum_{j=1}^{\kappa} Y_{t+h\Delta n} \quad (3.61)$$

where $h\Delta n$ is the prediction horizon at next instants $\{1, \dots, h\}$, $\{\alpha_j\}_{j=1}^{\kappa}$ are weights that add up to 1 expressed by Equation 3.62:

$$\sum_{j=1}^{\kappa} \alpha_j = 1 \quad (3.62)$$

There are two popular ways to assign the values to the weights α :

1. Set all the weights equally to predict the average value.

$$\alpha_j = \frac{1}{\kappa} \quad (3.63)$$

2. Set the weights inversely proportional to the closeness of the vectors \mathbf{D}_{v_k} respect with \mathbf{D}_t as described by Equation 3.64 and 3.65.

$$\alpha_j = \frac{\beta_j}{\sum_{i=1}^{\kappa} \beta_i} \quad (3.64)$$

where the partial weights $\{\beta_i, \dots, \beta_\kappa\}$ are assigned with Equation 3.65

$$\beta_j = \frac{\text{dist}(\mathbf{D}_t, \mathbf{D}_{v_\kappa}) - \text{dist}(\mathbf{D}_t, \mathbf{D}_{v_j})}{\text{dist}(\mathbf{D}_t, \mathbf{D}_{v_\kappa}) - \text{dist}(\mathbf{D}_t, \mathbf{D}_{v_1})} \quad (3.65)$$

where \mathbf{D}_{v_1} is the closest and \mathbf{D}_{v_j} is the farthest vector. As a result the closer delayed vector will have more influence for the prediction.

Besides of these methods for the weight assignment is possible to set the weights using other kind of kernels like gaussian, sigmoidal or another kind of nonlinear relation.

3.4.2 Qualitative Forecasting

When each element Y_t of the time series take elements $Y_t \in \mathbf{A}$ from any alphabet \mathbf{A} , where \mathbf{A} contains symbols used as labels, rather than scalar values, we call the time series qualitative. For example, classification labels for the identification of certain patterns associated with any class. Since these values are not related to any magnitude, the metric to measure the distance must considers only the qualitative differences of the elements of the delay vector \mathbf{D}_t . For this purpose, the Hamming distance is a suitable metric to measure the difference between two sequences of labels. The Hamming distance is shown in 3.66

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^m \text{hamming}(x_i, y_i)}{m} \quad (3.66)$$

where $\mathbf{x} = \{x_1, \dots, x_m\}$ and $\mathbf{y} = \{y_1, \dots, y_m\}$ are *label* vectors where $\mathbf{x}, \mathbf{y} \in \mathbf{A}^m$ are sequence labels of length m and the distance is evaluated as presented in Equation 3.67

$$\text{hamming}(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise.} \end{cases} \quad (3.67)$$

The estimation of the next h labels is performed by means of the statistical mode Mo (Equation 3.68)

$$\hat{Y}_{t+h\Delta n} = Mo(\{Y_{t+h\Delta n}\}_{j=1}^{\kappa}) \quad (3.68)$$

where $\hat{Y}_{t+h\Delta n}$ is the forecasted value at time t given by the mode of the set $\{Y_{t+h\Delta n}\}_{j=1}^{\kappa}$ produced by the nearest neighbours and $h\Delta n$ takes the prediction horizon values from 1 to h . This scheme is also known as majority vote, studied deeply by Bon Boyer and J. Strother Moore (BM91). The scheme also has important applications in multi-model classification systems ((Nip11), (SCM02) (Fre95)).

3.4.3 Simple Nonlinear Filter

A filter is a function that replaces a noisy measurement by a new value with less noise, and closer to the real value. In time series, filters are useful for removing noise in order to find a dynamic structure in a system. The Simple Nonlinear Filter presented in the book of Nonlinear time series analysis by H. Kantz (KS04), is an algorithm for noise reduction of nonlinear time series. This algorithm is based on finding similar patterns of any section of the time series. Once similar patterns are found, the average of these components is computed and the studied point is replaced by a new filtered point.

The raw data might contain noisy information in the time series. The recent raw data input is processed in real time and the filter should provide a new value with a reduced amplitude of the noise component. The noise filter tries to identify the data by decomposing it into two components: the signal structure and random fluctuations. This is presented as an additive superposition of the two components in Equation 3.69

$$Y_{t+1} = s_{sig,t+1} + \eta_{t+1} \quad (3.69)$$

where $s_{sig,t}$ is the noise-free signal and η_t is random noise with null autocorrelations.

In order to recover the signal s_t from the raw (and possibly noisy) signal Y_t , it is necessary the availability of a function \hat{f} that processes the data $\{Y_{t-m+1}, \dots, Y_t\}$. The ideal noise filter must satisfies Equation 3.70

$$s_{sig,t+1} - f(Y_{t-m+1}, \dots, Y_t) = 0 \quad (3.70)$$

which it is an equivalence approximated by replacing the t -th value by the middle $m/2$ average value of the nearest neighbours vectors. This function \hat{f} is given by Equation 3.71

$$\hat{s}_{sig,t_0-\lceil m/2 \rceil} = \frac{1}{|U_{\in}(Y_{t_0})|} \sum_{Y_t \in U_{\in}(Y_{t_0})} Y_{t-\lceil m/2 \rceil} \quad (3.71)$$

where $\hat{s}_{sig,t_0-\lceil m/2 \rceil}$ is the estimated *noiseless* value and $U_{\in}(Y_{t_0})$ is the set representing the neighbours close to Y_{t_0} .

3.5 Radial Basis Function Artificial Neural Networks

In machine learning, artificial neural networks (ANN) are computational models inspired in the biological neural networks applied to solve engineering problems as clustering, function modelling, regression and system identification for control (HH92), (SJ97). For the purpose of this thesis we are interested in the prediction capabilities of these kind of models.

A basic ANN is composed by an input layer of same size as the input vector, a hidden layer that weighs the result of applying a certain basis function to each element of the input vector, and the output vector is the weighted sum of the values from the hidden layer. Radial Basis Function Artificial Neural Networks (RBF-ANN) are a kind of artificial neural networks with the characteristic of using Radial Basis Functions in the neurons of its hidden layer (Orr96).

The Radial Basis Functions (RBF) are monotonically decreasing or increasing real-valued functions that depend just on the distance between the origin (or center) μ and another other point x . The center μ , the distance r and the radius or width σ the parameters of a RBF.

Therefore a RBF function $\psi(x, \mu)$ computes an associated value with the distance $r = \|x - \mu\|^2$. This value is the maximum when the distance, r , is the minimum, and the minimum value when r tends to infinity. This property makes that these kind of functions produce higher response for the values in the neighbourhood close to the centre.

Particularly, the RBF for the ANN considered in this work is the Gaussian Radial Basis that is one of the most used RBF functions for general purposes and it is given by Equation 3.72

$$\psi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mu_j\|^2}{2\sigma_j^2}\right) \quad (3.72)$$

where \mathbf{x} is the input vector, μ is the center of the function, and σ is the width of the exponential function. The mathematical model of the RBF-ANN can be expressed as follows

$$f_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \psi_j(\mathbf{x}) \quad (3.73)$$

where w_{kj} is the weight of the j -th neuron and the k -th output, M is the total number of hidden neurons, $f_k(\mathbf{x})$ is the k -th output layer of the neural network and ψ is a radial basis function. The weights \mathbf{w} in Equation 3.73 must be set to minimise Equation 3.74

$$\arg \min_{\mathbf{w}} \sum_{i=1}^N (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \quad (3.74)$$

where \mathbf{w} is the set of weights that minimises Equation 3.74 given the σ widths and μ centers. n is the total number of samples \mathbf{x} of the training set, $\hat{\mathbf{y}}_i$ is the mapping produced by the artificial neural network in Equation 3.73 and \mathbf{y}_i is the actual output associated with input \mathbf{x}_i .

3.6 Error Measurement Indicators

To measure the general performance of the error in forecasting models, different indicators are used in the literature. These indicators are useful metrics to compare the accuracy of forecasting models with the aim of avoiding the error cancelation produced by the accumulation of positive and negative errors. This is achieved removing the direction of the error either using the absolute value or squaring the error value. The i -th

error is an scalar value represented by $e(i)$. Popular error indicators or deviations used to summarize the error in a scalar vector are:

- **Mean Absolute Error (MAE):** Measures the average of the absolute errors of the error set with size n of forecasts. It can be interpreted as the average of all the equally weighted error magnitudes. This is expressed by Equation 3.75.

$$MAE = \frac{1}{n} \sum_{i=1}^n |e(i)| \quad (3.75)$$

- **Mean Squared Error (MSE):** It is a quadratic metric that computes the average of the individual square errors. This metric assigns higher weights to large magnitude errors with a quadratic proportion. It is expressed in Equation 3.76.

$$MSE = \frac{1}{n} \sum_{i=1}^n e(i)^2 \quad (3.76)$$

- **Mean absolute percentage error (MAPE):** This accuracy metric normalises the errors respect to the process mean giving the error in terms of percentage. This characteristic is useful to compare different performances in different time series belonging to different order. Since the re-escalation is applied, it makes the indicator a-dimensional. The definition is found in Equation 3.77.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{e(i)}{\mu} \right| \quad (3.77)$$

where μ is the mean of the full set of values.

- **Root Mean Square Error (RMSE):** It is a quadratic metric that measures square root of the average squared error. Since each error is squared, weights are assigned quadratically to the errors which they are then summed. The result of the sum is normalised when

it is divided by the number of elements n . It is useful for comparing vectors of predicted and observed values. This is expressed by Equation 3.78.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n e(i)^2}{n}} \quad (3.78)$$

The average $RMSE_h$ is a reference index for comparing the performance of the different forecasting algorithms at different prediction horizon values h . It is defined by Equation 3.79,

$$\overline{RMSE}_h = \frac{1}{n} \sum_{n'=1}^n \sqrt{\frac{\sum_{i=n'}^{n'+h-1} e(i)^2}{h}} \quad (3.79)$$

where n' is an auxiliary index that is moved one step ahead and then fixed to move the index i , h steps ahead. At each n' is computed the root square of the h next square errors $e(i)^2$ average. Finally the average of the independent vector errors is computed.

3.7 Classification and Clustering

This section addresses the k-means algorithm and the silhouette method used in data mining and machine learning for pattern recognition and identification. These methods are able to identify temporal behavior characteristics useful for local modelling which allow to build a multiple model able to capture and differentiate dynamical modes which present different statistical properties among them.

A clustering algorithm is a method that groups a set of vectors according to a specific similarity criterion (e.g, the euclidean distance, the infinity norm, or hamming distance). These algorithms work with the principle that vectors (or objects) may share characteristics in common so, the objects with similar characteristics are grouped in a class. In this way the clustering simplifies the description of the data diversity of a complex data set.

3.7.1 k-Means Clustering

The k-means algorithm is a data mining clustering method based on minimising the mean distance of each centroid to a set of selected data points according to their distances. The centroids represent different classes of data, and are used as prototypes. The k-means algorithm tries to minimise the objective function described as shown in Equation 3.80

$$\arg \min_{\{\mu_i\}_1^k} \sum_{i=1}^k \sum_{x \in \mathbf{x}} \|x - \mu_i\|^2 \quad (3.80)$$

where $\mathbf{x} = \{x_1, \dots, x_n\}$ is the set of objects to be classified, μ_i is the centroid to be optimised, and k is the number of clusters contained in the data set. Solving the k-means optimisation in Equation (3.80) is NP-hard but in practice it can be solved efficiently with the 2-step Lloyd's algorithm described by means of Equations (3.81) and (3.82). When the centroids are initialised (random initialisation is an accepted strategy), the next step is to update the next position of the centroids μ_i using Equation:

$$\mathcal{C}_i = \{\mathbf{x}_n : \|\mathbf{x}_n - \mu_i\| \leq \forall \|\mathbf{x}_n - \mu_l\|\} \quad (3.81)$$

where \mathcal{C}_i is the set of objects belonging to centroid μ_i , defined by the minimum distance given by $\|\mathbf{x}_n - \mu_i\|$ over all the centroids and the update is computed by Equation

$$\mu_i = \frac{1}{|\mathcal{C}_i|} \sum_{x \in \mathcal{C}_i} x \quad (3.82)$$

where μ_i is the updated centroid generated by the arithmetic mean of the objects in \mathcal{C}_i . In order to select the most suitable number of classes k that fit better the data, auxiliary methods to measure and validate the quality the data separation among classes might be used.

3.7.2 Selecting the Best Cluster Partition

In order to find the best separation of classes after the execution of a clustering algorithm, an index that indicates how well the clusters are

separated is used. Depending on the characteristics of the data the clustering algorithm for separating the data in groups with its number of clusters k . The parameter k is chosen from a set of possible values $K_{test} = \{1, \dots, k, \dots, k_{max}\}$. Once the clustering algorithm is executed, an indicator of quality Q_k is associated with the clustered data in $k \in K_{test}$ clusters. The criteria depends on the kind of index, it might be the maximum value of the indicator as expressed by Equation (3.83).

$$\arg \max_{k \in K_{test}} Q_k \quad (3.83)$$

Also the minimum value of the indicator as shown in Equation (3.84)

$$\arg \min_{k \in K_{test}} Q_k \quad (3.84)$$

or the $K_{test,k}$ that produces the biggest difference respecting $K_{test,k-1}$ as shown in Equation (3.85)

$$\arg \max_{k \in K_{test}} Q_k - Q_{k-1} \quad (3.85)$$

where Q_k is defined by Equation (3.86)

$$Q_k = Q(\mathcal{P}, k) \quad (3.86)$$

where \mathcal{P} is the partitioned data, k is the selected number of clusters in \mathcal{P} , and Q is the function that evaluates the index for \mathcal{P} using any algorithm. In the literature are found a plentiful of quantitative indexes for evaluating the clustering quality. A good survey with a good number clustering criteria is found in (Des13).

Silhouette Method

In clustering analysis, intuitive ideas are adopted for the development of clustering algorithms and indicators for measuring the cluster quality.

The concepts of cluster, mass, radius, density, cohesion, and separation are mixed and used for this proposal.

The silhouette method is an algorithm to compute a separability coefficient developed by Peter J. Rousseauw (Rou87) inspired by the human intuition of clustering and classification. The silhouette coefficient is reliable in compact and clearly separated clusters. The silhouette method requires the data set partitioned in k -clusters and the set of proximities between all the objects.

To measure the silhouette coefficient we define $s(i)$ as the dissimilarity distance of object i . A is the cluster to which object i belongs. The average similarity of i with the other j objects in A is computed and expressed as Equation 3.87 and 3.88.

$$a(i) = \frac{1}{|A| - 1} \sum_{j \in A \text{ and } i \neq j} dist(i, j) \quad (3.87)$$

$$b(i) = \text{minimum } dist(i, B) \quad (3.88)$$

where $dist(i, B)$ is the set of all similarity measures between i and objects $j \in B$. $b(i)$ is the minimum similarity measure of the nearest cluster $B \neq A$. With these definitions we are able to compute the individual silhouette coefficient for each object i using Equation 3.89.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.89)$$

Finally the average silhouette coefficient is given by Equation 3.90.

$$\frac{1}{n} \sum_{i=1}^n s(i) \quad (3.90)$$

In many real cases, there exists the possibility of having not perfectly separated clusters having mixed clusters, in this case the output of Equation 3.89 might be negative. The meaning of this is that there is an intrusive object inside the set A belonging actually to another cluster. For that reason $b(i)$ can be substituted by the mean of all the distances of the objects $i \in A$ with the objects $j \in B$.

$$b(i) = \frac{1}{|B| - 1} \sum_{j \in B \text{ and } i \neq j} dist(i, j) \quad (3.91)$$

where B is the closest cluster to the object i and $dist(i, B)$ is the set of dissimilarity distances between i and the j objects contained in B .

3.7.3 Feature Extraction

Since the proposed Multi-Model Predictor addressed in Chapter 4 is based strongly on the feature extraction for the identification of patterns, this subsection includes a general guide introducing a general classification of different feature extraction algorithms that can be implemented for the identification and modelling of dynamical patterns. The feature extraction is a kind of data processing to obtain meaningful characteristics from the raw data in order to improve the performance of algorithms oriented to system modelling and identification. This processing is useful to remove redundant data, select local features or enhance a signal. A classification of the feature extraction algorithms is shown next.

- **Standardisation and normalisation:** These algorithms translate to a common scale to compare and analyse data from different sources with different scales. In time series it is convenient to scale the data sequence to unique range in order to compare the performance of different forecasting methods applied to different time series with different range. These algorithms are studied widely and applied in machine learning and pattern recognition (SKF08), (AH01).
- **Signal enhancement:** In the nature regularly the signal is noisy or weak to manipulate. This is because the signal to noise ratio is high. In order to reinforce the signal, it is possible to apply denoising, smoothing, sharpening or de-trending filters. As example, differentiating is a simple de-trending operation that removes the trend from the time series in order to obtain a stable mean in time series. This is useful when is required to compare the qualitative characteristics of the time series. This algorithms are also found in instrumentation research (ELW⁺08), (Ras14).

- **Extraction of local features:** Methods for encoding the data and transform the knowledge in features. In time series this consists on finding the proper model for certain sections of the time series with specific characteristics. The local features are modelled by independent models selected following certain rules and conditions. A survey about time series clustering by using local features is found in (War05).
- **Linear and nonlinear space embedding methods:** Since the Multi-Model Predictor framework is based on clustering and regression methods, they may have limitations dealing with high dimensional data. In order to deal with this fact, it is important to discard the irrelevant components that do not contribute with relevant information. Data discrimination is useful for two reasons: The number of optimisation variables might be reduced and the computational performance might be improved once the irrelevant data is not computationally analysed anymore. For example, old past information of time series with negligible influence in the recent dynamics of the system is better to be discarded. Another example in nonlinear time series, is the use of mutual information and false neighbours algorithms (KS04) for detecting and discarding useless information for system modelling.
- **Nonlinear expansion:** When the data is too complex, is common to transform the data into a higher dimension. This idea is applied to nonlinear time series, where observing just one variable in a unidimensional space, is possible to reconstruct the trajectory in a higher dimensional space. The theory of machine learning classifiers as ANN and SVM provides the possibility of implementing a set of hidden neurons with nonlinear activation functions to transform or map the input data in a nonlinear space. For the Support Vector Machines the implementation of kernel tricks is useful also for transforming the dimension and shape of the original feature space of the data to another that improves the performance. Examples of applications exploring the nonlinear relationship of the natural

data are found in (CLX⁺11) and (She05).

- **Feature discretisation:** For dealing with computational complexity, some algorithms require to discretise the continuous model or data. For example, for the implementation of a dynamical system for simulation in MPC for being solved by computers or micro controllers since the continuous characterisation of the model is intractable for computers. An important article about discretisation of partial differential equations and discretisation of continuous attributes as preprocessing step for machine learning are found in (BWZ08) and (CGB96)

Here is concluded the basic theory and the description of the algorithms involved to address the Multi-Model Predictor framework. The Multi-Model Predictor Framework and three implementations are addressed and described in detail in Chapter 4.

Chapter 4

Multi-Model Forecasting

Time series can present several pattern behaviours in regular time lapses activated by events generated by natural or social circumstances (e.g. social events, meteorological phenomena, regime of chemical reactions). This is, for example, the case of water or electricity consumption that presents a daily periodicity (the demands present a repetitive pattern every day varying regularly during week-ends and holidays), a weekly periodicity (the demand decreases during the week-ends) and seasonal changes (the demand changes according to the season because of weather, holidays, etc.).

Several approaches have been proposed in the literature for forecasting time series that presents several patterns and seasonalities, see e.g., (AFM07) or (QPG⁺10), where a two-level model is used: a daily flow time series model is combined with a daily 10-minute distribution pattern. The identification of these patterns allows to design local models to forecast specific regimes. Recent related research on identifying behavior regimes in time series can be found in the literature. In (BC13), the implementation of a global NAR neural network predictor is proposed to estimate the regimes associated with local NAR neural network predictor used for forecasting the hourly global solar radiation. In (KP10), an algorithm is proposed for clustering the data and training local predictive models for each set to generate a forecast based on the combination of the

local models. In (MALS⁺07), clustering is used to group the days with similar pattern with regard to the variation of working days and holidays. In fact, time series forecasting presenting multiple patterns is related to the multi-modelling approach of dynamical systems (SJ97). This is an approach used for modelling nonlinear systems under the assumption that they can be approximated by a finite number of interpolated linear invariant (LTI) models in different operating points (regimes). The identification of those multi-models also involves identifying the different regimes. Once identified, a linear model is estimated for each regime and these local models are combined on-line using weighted sum and a scheduling variable.

This Chapter describes three different implementations of the Multi-Model Forecasting Framework. The Section 4.1 describes the general predictor architecture that they use. The three algorithms are classified in two groups according to their similarity in their off-line (preparation) and on-line (operational) phases:

1. Qualitative and Quantitative Multi-Model Predictor **QMMP** that includes:
 - (a) Multi-Model Predictor based on the Qualitative and Quantitative Decomposition of time series using SARIMA and kNN as qualitative estimator (**QMMP+kNN**).
 - (b) Multi-Model Predictor based on the Qualitative and Quantitative decomposition of time series using SARIMA with a noise filter using a predefined calendar activity as qualitative estimator(**QMMP+NF**).

These algorithms are based on the qualitative and quantitative decomposition which are inspired on the SARIMA pattern algorithm proposed by Quevedo in (QPG⁺10) (**QMMP**).

2. ANN Multi-Model Predictor

- (a) Multi-Model Forecasting using Radial Basis Function Neural Networks with an on-line Mode Recognition (**RBFMMP+OR**)

that consists in a framework for training machine learning Multi-Model predictors.

The Multi-Model Forecasting **RBFMMP+OR** algorithm is described in Section 4.2. A particular case is addressed using a collection of algorithms that allows the multi-model prediction, such as k-means for pattern classification, RBF-ANN for the Multi-Model Predictor, and a mode recognition mechanism based on the nearest neighbour rule.

In Section 4.3, the two Multi-Model Predictors based on the decomposition of the quantitative and qualitative decomposition of time series **QMMP** are described. The first configuration **QMMP+kNN** implements two parallel forecasters where the qualitative predictor assumes the lack of future information to activate the most suitable operation mode. For this implementation, the decisions are based completely on the history of the operation modes sequence. The second implementation, assumes the existence of an activity calendar with the definition of the sequence of working and resting days related to the forecasting operation modes. This scheme is extended with a nonlinear filter module for modelling and predicting the mismatches to improve the prediction accuracy. Both algorithms use a low order SARIMA model to forecast the quantitative (aggregated) component of the time series.

4.1 General Predictor Architecture

The proposed general architecture is designed, as we mentioned before, to be used as general modelling methodology applicable to time series and also to systems presenting multiple behaviours. The proposed framework provides a methodology for:

- The preparation phase consisting of the off-line training of the **Multi-Model Predictor (MMP)** and
- The execution phase that is the on-line implementation of the **Multi-Model Predictor (MMP)**.

The Multi-Model Predictor **MMP** has an architecture composed of interconnected modules involved in the mode recognition or prediction, and activation of the suitable model predictor. This architecture is described in Figure 18. It has three main modules: **Feature Extraction**, **Mode Recognition** and the **Multi Forecasting Model**.

The function provided by the **Feature Extraction** module allows to transform the data to extract qualitative information from the time series (GE06). This is done by transforming the time series input into suitable qualitative representation extracting detrended unitary patterns that can be used by the **Mode Recognition**, and into quantitative representation to be used by the Multi Forecasting Model. The **Mode Recognition** module estimates the next pattern to be activated using the processed information produced by the **Feature Extraction**, and the **Multi-Model** module contains a set of independent forecasting models associated with each mode.

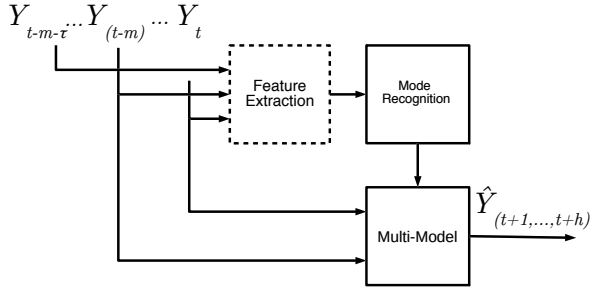


Figure 18: General structure of the Multi-Model Predictor

4.2 Multi-Model Forecasting Using RBF-ANN with an On-line Mode Recognition

The proposed algorithm based on machine learning algorithms allows the off-line identification and training of the different dynamic behaviours presented in time series, and the on-line model recognition and activa-

tion according to the detected pattern. A particular implementation based on k-means and Radial Basis Functions (RBF) is proposed to exemplify a possible implementation oriented to the short term water demand forecast using demand data from the Barcelona drinking water network.

The Multi-Model Predictor **MMP** consists in a sequence of steps that involves the training of a machine learning model using classified patterns of the time series. This process is described by means of the block diagram of the Figure 19. The raw data might be required to be pre-processed to enhance the mode and cluster identifiability of the patterns considering the **Feature Extraction** as an optional component.

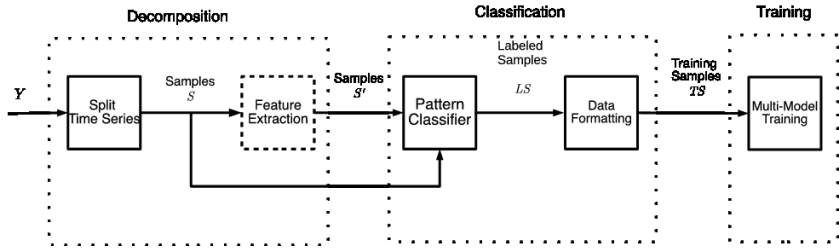


Figure 19: Multi-Model training process

The steps involved in the off-line **MMP** training are:

1. **Decomposition**: The training set is organised in samples S of the time series given by the **Split** Module. If needed, the output of this module is transformed to another feature space by the means of **Feature Extraction** module responsible of mapping the vectors S into vector S' according to Equation 4.1.

$$\Psi : S \mapsto S' \quad (4.1)$$

2. **Classification**: This step includes the **Pattern Classifier** module for classifying the split time series S (if the feature spaces is transformed, then the classification is performed with the transformed vectors S') extracted by the **Feature Extraction** module in order to generate labeled samples LS that include the assigned class. The

classified labeled samples LS are formatted by the **Data Formatting** module that generates training sample vectors TS that contain input-output training information with the associated class attached.

3. **Training:** In this step, a machine learning model is trained using the input-output samples TS .

Since the proposed framework might be seen as an open modular model, there exists a large collection of machine learning and pattern recognition algorithms in the literature that can be used for the implementation of the proposed **MMP**. However, only a particular implementation using standard and well known algorithms used in machine learning literature, is addressed with the objective of illustrating the improvement of the prediction accuracy over the traditional methodologies (JV08) - (XKQ⁺08). Therefore the following algorithms are used:

- k-means clustering algorithm for the **Pattern Classifier** module,
- RBF neural networks for the **Multi-Model** predictor module, and
- Discrete differentiation of time series for the **Feature Extraction** module.

The implementation of the **Split Time Series**, **Data Formatting** and **Mode Recognition** modules are addressed in the next section as part of the training step.

4.2.1 Discrete Derivative as a Feature Extraction Method

The **Feature Extraction** module performs the task of transforming raw data into a set of useful features (GE06). The function of the feature extraction is to improve the classification performance of the data. The discrete derivative (or differentiation) of the time series is implemented as feature extraction method described by

$$Y'_t = Y_t - Y_{t-1} \quad (4.2)$$

where Y'_t is given by the difference of the data at time t at lag $t - 1$.

The effect of this transformation consists in removing the mean and trend taking the relative increments of the time series used for the mode identification (MG08).

4.2.2 On-line Mode Recognition for the Multi-Model Predictor Approach

The proposed methodology for building the Multi-Model Predictor is designed to use machine learning regression methods that learn a model that associates the input with the output vectors from a subset of samples (AYK⁺11). In order to build the Multi-Model Predictor, the specific algorithms for the **MMP** training and On-line Mode Recognition with the Multi-Model Predictor (**RBFMMP+OR**) implementation are described next.

Preparation Phase

In the preparation phase is performed the training of the **Multi-Model** predictor. This phase involves the **Decomposition**, **Identification** and **Training** steps performed off-line. The implementation of each step involved is described next.

Decomposition:

This step decomposes the time series for the identification of the different behavior patterns. The decomposition starts assuming the availability of an univariate time series described by Equation 4.3,

$$\mathbf{Y} = \{Y_1, \dots, Y_i, \dots, Y_n\} \quad (4.3)$$

where Y_1 , Y_i and Y_n are the first, the i -th and the last element of the time series respectively. The time series \mathbf{Y} is decomposed in samples \mathbf{S} as presented in Equation 4.4.

$$\mathbf{S} = \{S_1, \dots, S_i, \dots, S_{N'}\} \quad (4.4)$$

where N' is the total number of samples defined by $\lfloor \frac{n-h-m+1}{\tau} \rfloor$ and S_i is the sample obtained splitting the original time series \mathbf{Y} according to Equation 20

$$S_i = \{Y_{(\tau \cdot i - m + 1)}, \dots, Y_{\tau \cdot i}, \dots, Y_{\tau \cdot i + h}\} \quad (4.5)$$

where m is the number of observations, h is the desired prediction horizon, and τ is the period parameter that defines a split point that is applied every τ number of displacements. The τ displacements are performed by a sliding window of size $m + h$ inside each sample S_i in order to construct the training samples TS . A graphical description of the data organisation used for the construction of the training set is shown in the Figure 20, where S_1 and S_2 are samples of size $m + h + \tau - 1$ that allows τ displacements of the sliding window inside each sample.

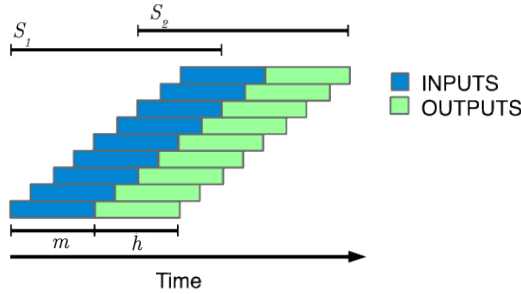


Figure 20: Time Series processing data

Identification and training

The objective of the identification analysis is to detect the number of different modes in the samples \mathbf{S} . The samples \mathbf{S} are classified into a suitable number of classes k using a clustering algorithm to group them according to their similarity. Then, a local forecast model is trained with classified data for each specific mode. In order to choose a number of clusters k , the classification is validated by maximizing the mean silhouette coefficient shown in Equation 3.89. When the **Feature Extraction**

module is implemented, the **Pattern Classifier** uses the samples S' that are transformed in the same feature space to classify S . After the classification of S , labeled samples $LS_i = \{C_i, S_i\}$ are obtained where $C_i \in K$ are the possible labels of C_i with $K = \{1, \dots, k\}$. Then, the class information is incorporated using extended input vectors that include the m last observations concatenated with the codified class information. The procedure for generating training samples **TS** is implemented in the **Data Formatting** module. To generate the training samples **TS**, a sliding window of size $m + h$ is used to obtain from the labeled samples **LS** the training samples $\mathbf{TS} = \{\{C', \mathbf{Input}\}, \mathbf{y}\}$ where $C'_i = \{\mathbb{I}_{C_i,1}, \dots, \mathbb{I}_{C_i,k}\}$ is the codification of the class in c_i using the $C_i - th$ row of the identity matrix $\mathbb{I}_{k \times k}$ as shown in:

$$\begin{bmatrix} k_1 \rightarrow k'_1 \\ \vdots \\ k_k \rightarrow k'_k \end{bmatrix} = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}_{k,k} = \mathbb{I}_{k,k} \quad (4.6)$$

The vector $Input_i = \{S_{i,j}, \dots, S_{i,j+m-1}\}$ is a sequence of m inputs and $\mathbf{y}_i = \{S_{i,j+m}, \dots, S_{i,j+m+h}\}$ is the output or target vector of size h where $j = \{1, \dots, \tau\}$ is the index that moves the sliding window τ times composing the training set by extending the each input vector to $\mathbf{x}_i = \{C_i, Input_i\}$ obtaining a set of training samples denoted by $\mathbf{TS} = \{\mathbf{x}_i, \mathbf{y}_i\}_i^{N'}$.

Training

The machine learning method is trained using the training samples **TS** to optimize the objective function

$$\underset{\mathbf{w}}{\text{minimise}} \sum_{i=1}^{N'} (f(\mathbf{x}_i) - \mathbf{y}_i)^2 \quad (4.7)$$

where \mathbf{w} is the set of weights that minimises the sum of the squared errors between the estimated output by the neural network denoted by $f(\mathbf{x}_i)$ and the real output vectors \mathbf{y}_i .

Operational Phase

The operational phase consists in the implementation of the execution the Multi-Model Predictor. After training the RBF-ANN, arises the challenge of designing a mechanism for the **Mode Recognition** module to estimate and discover the current operation mode. For this module, a mode discovery based on the nearest neighbour rule using a variable queue of observations is proposed. The idea is based on the exploitation of the current and limited historical information to estimate the class associated with class of the observed measurements using the Euclidean distance. The implementation consists in a queue \mathcal{W}' of variable size that stores from m to $m + \tau - 1$ measurements as follows in Equation 4.8,

$$\begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \vdots \\ \mathcal{W}_{m+mod(t,\tau)} \end{bmatrix} = \begin{bmatrix} Y_{t-(m+mod(t,\tau))+1} \\ Y_{t-(m+mod(t,\tau))+2} \\ \vdots \\ Y_t \end{bmatrix} \quad (4.8)$$

where \mathcal{W}_1 is the first element of the queue that stores a delayed measurement $Y_{t-(m+mod(t,\tau))+1}$, and $\mathcal{W}_{m+mod(t,\tau)}$ stores the current measurement Y_t . Notice that \mathcal{W}_1 is realigned each period defined by τ . In order to simplify the notation, we define in Equations 4.9 and 4.10, the queue $\{\mathcal{W}'\}^{sup(t)}$ and the prototype vectors $\{\mathbf{P}_k\}^{sup(t)}$.

$$\{\mathcal{W}'\}^{sup(t)} = \{\mathcal{W}_1, \dots, \mathcal{W}_{sup(t)}\}, \quad (4.9)$$

$$\{\mathbf{P}_k\}^{sup(t)} = \{P_1^k, \dots, P_{sup(t)}^k\}, \quad (4.10)$$

where $sup(t) = m + mod(t, \tau)$. The estimation of the class associated with the current observations in time t stored in $\{\mathcal{W}'\}^{sup(t)}$ is performed using the nearest neighbour rule defined by Equation 4.11

$$\arg \min_{k \in K} \left\| \{\mathbf{P}_k\}^{sup(t)} - \{\mathbf{Z}\}^{sup(t)} \right\| \quad (4.11)$$

where the class selected is the argument k that minimises the distance between the prototype $\{\mathbf{P}_k\}^{sup(t)}$ and the current measurement stored in $\{\mathcal{W}'\}^{sup(t)}$.

4.3 Multi-Model Predictor Based on Qualitative and Quantitative Decomposition

In this Section, two implementations of the Multi-Model Predictor are addressed exploiting the decomposition of the time series in its qualitative and quantitative components; a Multi-Model Predictor with qualitative forecasting assuming no knowledge about the future (e.g., the lack of a calendar activity), and another implementation assuming the calendar of activity that provides the forecasting modes but including a nonlinear filter for detecting mismatches in the qualitative pattern prediction. The architecture of the Multi-Model Predictor is described previously in Figure 18. In both implementations are included:

- A seasonal ARIMA model that predicts the quantitative component using daily aggregated magnitudes.
- A mode estimator that can be implemented using different strategies depending on the knowledge data assumption. In the first case, it is assumed that the sequence of the modes are not known a priori and in the second case it is assumed that the operation modes are a priori known as a good approximation of the global behavior, for example, the annual activity calendar of working and resting days governing the behavior of the urban population. The modelling study of such information belongs to the global modelling.
- A k-means clustering algorithm for finding the patterns modes.

To perform the training and tuning parameters of the different algorithms involved, the raw time series described by Equation 4.3, is decomposed in its qualitative and quantitative components. The quantitative time series is given by Equation 4.12

$$Z_i = \sum_{j=((i-1)\tau+1)}^{i\cdot\tau} Y_j \quad (4.12)$$

where τ indicates the seasonality while the qualitative (or normalised) time series is obtained by Equation 4.13:

$$X_i = \frac{\{Y_{((i-1)\tau+1)}, \dots, Y_{i\cdot\tau}\}}{Z_i} \quad (4.13)$$

where $\mathbf{X} = \{X_1, \dots, X_i, \dots, X_N\}$ is the time series containing the adimensional qualitative unitary patterns. The qualitative patterns are classified and associated with a class $K \in \{1, \dots, k\}$. The associated classes are contained in \mathbf{C} defined by Equation 4.14

$$C_T = \arg \min_{i \in K} \|X_T - P_i\| \quad (4.14)$$

where P_i is the distribution pattern or prototype, given by the mean of the objects contained in each class, and k is the number of classes. Each object $C_T \in K$ contains a label from K . The new time series with the sequence of labels C_T is defined in Equation 4.15.

$$\mathbf{C} = \{C_1, \dots, C_T, \dots, C_N\} \quad (4.15)$$

Time series prediction models for \mathbf{Z} and \mathbf{C} are modelled independently. On one hand, the estimation of the next value \hat{Z}_{T+1} might be performed by a low order Seasonal ARIMA. On the other hand, \hat{C}_{T+1} can be estimated using two main general approaches based on the certainty assumption of the data. The first one is using the a priori knowledge of a human based calendar assuming that the patterns approximate the activity calendar of working and resting days. The second approach consists in using a general method for determining the next pattern automatically without any assumption. For this latter approach, methods based on the knowledge representation as kNN, Bayesian Networks, Markov Chains can be used.

The final forecasting is produced by the composition of the predicted qualitative and quantitative components $\mathbb{Z}_T = \{\hat{Z}_{T+1}, \dots, \hat{Z}_{T+H}\}$ and $\mathbb{P}_T = \{P_{\hat{C}_{T+1}}, \dots, P_{\hat{C}_{T+H}}\}$, as follows

$$\begin{aligned} \hat{Y}_{(T\tau)+1}, \dots, \hat{Y}_{(T+1)\tau} &= \hat{Z}_{T+1} \cdot P_{\hat{C}_{T+1}} \\ &\vdots \\ \hat{Y}_{(T+H-1)\tau+1}, \dots, \hat{Y}_{(T+H)\tau} &= \hat{Z}_{T+H} \cdot P_{\hat{C}_{T+H}} \end{aligned} \quad (4.16)$$

where τ is the length of the period, H and T are the prediction horizon and current time respectively in a τ period basis.

4.3.1 Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition of Time Series using SARIMA and kNN

The implementation of this Multi-Model Predictor with a Mode Prediction without any information about the future of the prediction mode, uses a qualitative forecasting model based on kNN. The scheme is described in Figure 21.

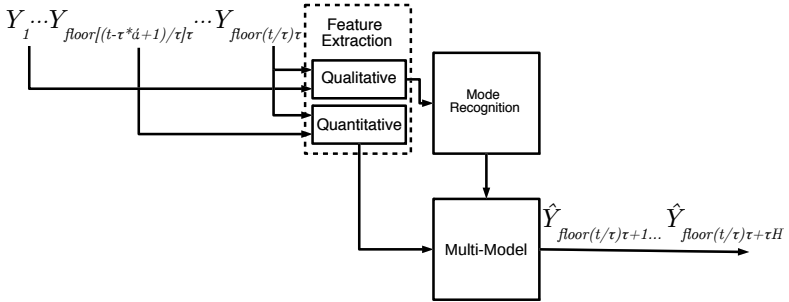


Figure 21: Multi-model training process

According to Figure 21, the observed data so far $\{Y_1, \dots, Y_{\text{floor}(t/\tau)\tau}\}$ is received and processed by the **Qualitative Feature Extraction** module

delivering the extracted qualitative information to the **Mode Recognition** module. On the other hand, the subset of observations

$$Y_{floor[(t-\tau*\alpha+1)/\tau]\tau}, \dots, Y_{floor(t/\tau)\tau}, \quad (4.17)$$

(where α constraints the last days to be considered) is processed by the **Quantitative Feature Extraction** module related directly with the seasonality value used by SARIMA forecasting model. The SARIMA forecasting model is included in the **Multi-Model** module. The implementation is composed by the preparation and operational phases using data organised in training and validation sets respectively. The procedures of both phases are described next.

Preparation Phase

For the preparation phase, the parameters of the qualitative kNN predictor and SARIMA model are tuned using the training data set. The parameters ε and delays m of the qualitative kNN forecasting are found in Equation 3.66, 3.67 and 3.68 in Subsection 3.4.2.

The correlation analysis is required to identify the SARIMA model, and then optimise its parameters with any gradient based parameter estimation algorithm. Once the SARIMA model is obtained, the kNN parameters are adjusted executing exhaustive search bounding the search space by means of the parameters ε and m minimising the \overline{RMSE}_h errors as follows in Equation 4.18:

$$\arg \min_{m, \varepsilon} \sum_{T=1}^N \|\hat{\mathbf{Y}}_{T\tau+1} - \mathbf{Y}_{T\tau+1}\| \quad (4.18)$$

where $\hat{\mathbf{Y}}_{T\tau+1}$ is the predicted value given by the composition of the qualitative and quantitative forecasting for the next days expressed by Equation 4.19

$$\hat{\mathbf{Y}}_{T\tau+1} = \{\hat{Z}_{T+1}P_{\hat{C}_{T+1}}, \dots, \hat{Z}_{T+H}P_{\hat{C}_{T+H}}\} \quad (4.19)$$

and $\mathbf{Y}_{T\tau+1}$ is the real hourly information (in our application, water consumption) defined by

$$\mathbf{Y}_{T\tau+1} = \{Y_{T\tau+1}, \dots, Y_{\tau(T+H)}\} \quad (4.20)$$

where H is the H -th predicted vector with τ elements. The Figure 22 shows how the prediction at the smallest basis (hours) is performed along the period (day), using a sliding window of size h covering the prediction subsequence data from $(\tau T) + 1$ to $(\tau(T + H))$ in τ steps.

Operational Phase

In the operational phase, the Multi-Model Predictor architecture presented in the Figure 21 is implemented. In real-time, the model collects the raw measurements as input for transforming the data to allow the extraction and classification of the qualitative patterns performed by the **Qualitative Feature Extractions** and the aggregated data by the **Quantitative Feature Extraction** every τ steps. The sequence of labels C (Equation 4.15) are processed by the **Mode Detection** Module to estimate with Hamming distance based kNN, the next pattern to be considered and used by the **Multi-Model** Module.

For the operational control of water distribution networks, it is necessary to have available 24 hour ahead predictions not only in each period τ , but in an hourly basis. For expanding the capability of having the prediction information, kNN estimates two modes ahead, and SARIMA predicts two aggregated consumption days described as the Figure 22. The prediction at any time t is provided by a sliding window over $\hat{\mathbf{Y}}_{t+1}$ shown in Figure 22 with a desired prediction horizon h .

4.3.2 Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition of Time Series Using SARIMA and Noise Filter

This is another implementation of the Multi-Model Predictor using a low order SARIMA and an human activity calendar based on the information given by a human expert with qualitative sequence definition of resting and working days.

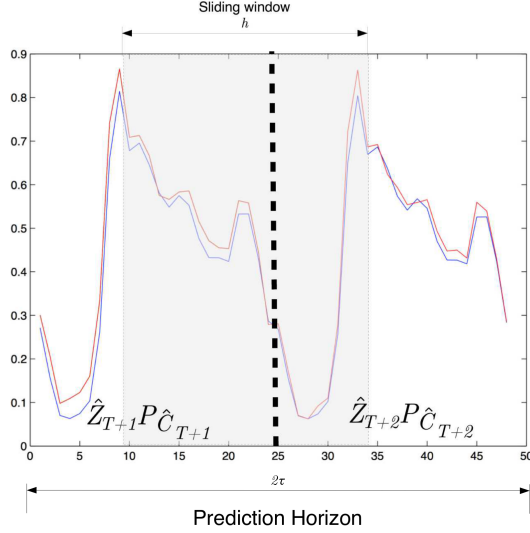


Figure 22: Prediction sample

The contribution of this algorithm is the introduction of a noise filtering module to address the problem of filtering the residuals or (mismatches) generated by the Multi-Model Forecasting (**MMP**) proposed by (QSPB14) in short-term Demand Forecasting for Operational Control and generalised in (LPR15). The objective of filtering the residuals is to discover the structure that improves the accuracy of the 24 hours ahead water demand forecasting for operational control. The algorithm uses a simple nonlinear filter to separate the noise from the structure of the residuals. The extracted information is useful for correcting the prediction in the next time instant.

In linear systems, the residuals of a good model fit a normal distribution with zero mean and constant variance along time. Real time series, e.g., those extracted from nature, might be influenced by external dynamics, changing gradually the behavior and characteristics of the system. Under those conditions, we need to readjust the parameters once the prediction residual structures or the increasing of the error predic-

tion are noticed, or even worst, use a new model to fit the new behavior properly. For cases where the systems present *sporadic* unmodelled dynamics, the mismatch must be processed and predicted to improve the accuracy or the forecast.

Implementation

The **Filter** module is an extension of the the **Multi-Model Predictor** architecture. We call it **Qualitative Multi-Model Predictor with Noise Filter (QMMP+NF)**. The **Filter** module provides a noise filtering mechanism to improve the forecasting accuracy and it is included in Figure 23.

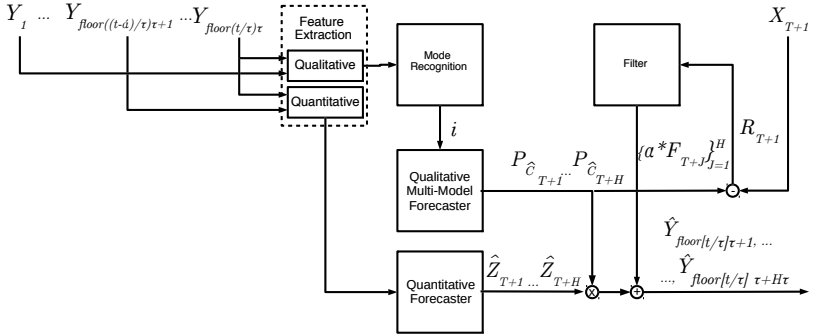


Figure 23: Multi-Model Predictor with the Filter Module

The **Multi-Model** module is seen as separated **Qualitative Multi-Model Forecaster** and **Quantitative Forecaster** modules. The implementation of the proposed methodology is divided also in the preparation and operational phase. In the preparation phase the separation of the data in training, validation and test sets is performed and the implementation of all the modules. In the operational phase the prediction mechanism for having the forecasting available for each hour along the day. The description of the two phases is explained in the next subsections is implemented.

Preparation Phase

The description of the modules of the **QMMP+NF** are:

- **Feature Extraction:** It performs the decomposition of the time series in its qualitative and quantitative components **X** and **Z** by using the Equations (4.12) and (4.13).
- **Mode Recognition:** This module is based completely on the knowledge of the human activity. It is represented by the function \hat{F}_{cal} that is the two class calendar function defined a priori according to the working and resting days, so the next mode is predefined by the calendar defined in Equation (4.21)

$$\hat{C}_{T+1} = \hat{F}_{cal}(C_1, \dots, C_i, \dots, C_T) \quad (4.21)$$

where each element $C_i \in [1, \dots, k]$ contains the class label associated with the pattern mode P . The validation of the calendar activity presented in the time series is performed by k-means and the silhouette coefficient.

- **Qualitative Multi-Model Forecaster:** This module receives the next mode selected by the strategy implemented in the **Mode Recognition** module. It contains the different patterns associated with different classes.
- **Quantitative Forecaster:** It includes a SARIMA model using the quantitative time series **Z**. It is tuned with the training set and validated with the validation set.
- **Filter:** This module is tuned once the previous modules are defined and implemented. The **Filter** collects the residual vectors **R** produced by the difference of predicted qualitative component $P_{\hat{C}_{T+1}}$ and the real qualitative component in X_{T+1} , producing the residuals.

$$R_{T+1} = P_{\hat{C}_{T+1}} - X_{T+1} \quad (4.22)$$

Filtering is performed on current time T , H steps ahead. It uses the residuals stored in $\mathbf{R} = \{R_1, \dots, R_T\}$. Once the predictions are available, the residual indexes associated with the modes $\{\hat{C}_{T+1}, \dots, \hat{C}_{T+H}\}$ are collected in vector V following the Equation (4.23)

$$V = \{\forall i \in [1, T], C_i = \hat{C}_{T+1}\} \quad (4.23)$$

The most recent occurrence of the mode \hat{C}_{T+1} in the past is taken using Equation (4.24).

$$s = \max(V) \quad (4.24)$$

where \max function gets the vector element with the maximum value and it is stored in s . The last occurrence of the predicted mode \hat{C}_{T+1} is used as reference for searching κ similar residuals to R_s . The residual R_s is selected as the basis for searching and sorting its κ similar residuals. In Equation (4.25) the vectors R_{V_l} are sorted and then taken the κ closest residuals to R_s

$$\text{dist}(R_s, R_{V_l}) < \text{dist}(R_s, R_{V_{l+1}}) \quad (4.25)$$

where function dist is the Euclidean distance function and $l = \{1, \dots, \kappa - 1\}$ are the indexes related to the closets residuals. Once V indexes are ordered, the filter for the next time step $T + 1$ is computed by the following equation (that is the simplification of Equation (3.71) setting $m = 1$).

$$F_{T+1} = \frac{1}{k} \sum_{i=1}^{\kappa} R_{v_i} \quad (4.26)$$

The F_{T+1} vector is used as predictor corrector and it is integrated in Equation (4.27).

$$\{\hat{Y}_{T\tau+i}\}_{i=1}^{H\tau} = \{\hat{\mathbf{Z}}_{T+i} \cdot \mathbf{P}_{\hat{C}_{T+i}} + \alpha F_{T+i}\}_{i=1}^H \quad (4.27)$$

where $\alpha \leq 1$ is a real constant number that weighs the amplitude of the filter signal. Notice that when $\alpha = 0$ the filter has no effect, and when $\alpha = 1$ the filter might over re-adjust the prediction producing poor

accuracy forecast. To estimate a value for the α parameter, the **MMP+NF** is executed with different α and κ values using the validation set and are selected those that minimize the prediction error.

Operational Phase

For the operational phase of water distribution networks, as it is mentioned before, it is necessary to have available 24 hour ahead predictions in an hourly basis for the operational control. Similarly to the previous algorithm, two days ahead are predicted and a window of size τ is slid over the prediction $\hat{Y}_{T\tau+1}, \dots, \hat{Y}_{\tau(T+H)}$. This process is described by Figure 22. To measure the performance of this architecture in an hourly basis is obtained using the testing data set.

With this last algorithm is concluded this Chapter. Next, the evaluation performance is addressed in Chapter 5 implementing the proposed forecasting models to predict the drinking water demand.

Chapter 5

Results

This chapter addresses the performance comparison in the short term of the proposed multi-model forecasting algorithms. In Section 5.1 the database of water demand times series is described as well as their general characteristics. Section 5.2 shows the experiments and validation for the proposed multi-model algorithms.

5.1 Description of the Database

To illustrate application with real data of the proposed forecasting models, a brief description of the time series origin is explained. The time series are selected from the database generated by the flowmeters of the water supply network (WSN) of Barcelona. The WSN of Barcelona is managed by AGBAR company (*Aguas de Barcelona*). AGBAR supplies water to Barcelona and metropolitan area. The water delivered comes from different water resources: from the rivers Ter and Llobregat, the underground flows and wells that provide water through pumping stations. The total water resources provides a flow approximately of $7m^3/s$. A desalinisation plant is also part of the DWN located at the end of Llobregat river, it produces drinking water by treating the sea water. This plant of $60hm^3/year$ capacity, is strategically convenient specially in case of drought periods.

The water demand often presents a pattern repeated daily with variations during weekends, and also, sometimes between winter and summer.

The experiments are performed with measurements taken by flowmeters at the entrance of the demand sectors that present a typical behavior of water consumption of residential zones. Since the database is composed by raw time series directly acquired from the flowmeters, some time series present missing or irregular data due to technical issues such as poor transmission quality or malfunction. Due to this, the data should be validated or preprocessed before being used as correct data. The validation and reconstruction of raw data is a task that is out of the scope for this thesis, nevertheless it is possible to make a preliminary outlier analysis with well known statistical tools. We detect and count the outliers adopting the modified Thompson *tau* technique (DI07). Thompson *tau* helps to decide whether to keep or discard suspected outliers in a sample of a single variable. The sequences considered for the analysis are those with zero or few outliers. The Figure 24 and 25 show the plots of all the complete time series found in the database with the corresponding number of detected outliers. The sequences remarked in red are considered for testing the proposed multi-model forecasting methods. The vertical axis is the normalised magnitude Y and the horizontal axis represents the time t in hours.

The time series are normalised transforming the original range values to the bounded domain $[0, 1]$. The flowmeters are enumerated to identify the sector that they cover with their corresponding detected outliers. We choose those time series 5, 11, 14, 20, 78 and 90 containing less than 70 outliers shown in Table 2, and one time series with an irregular data segment enumerated as 19. The sectors correspond to those with code in the original database p10007, p10015, p10017, 10026, p10095, p10109 and p10025 respectively.

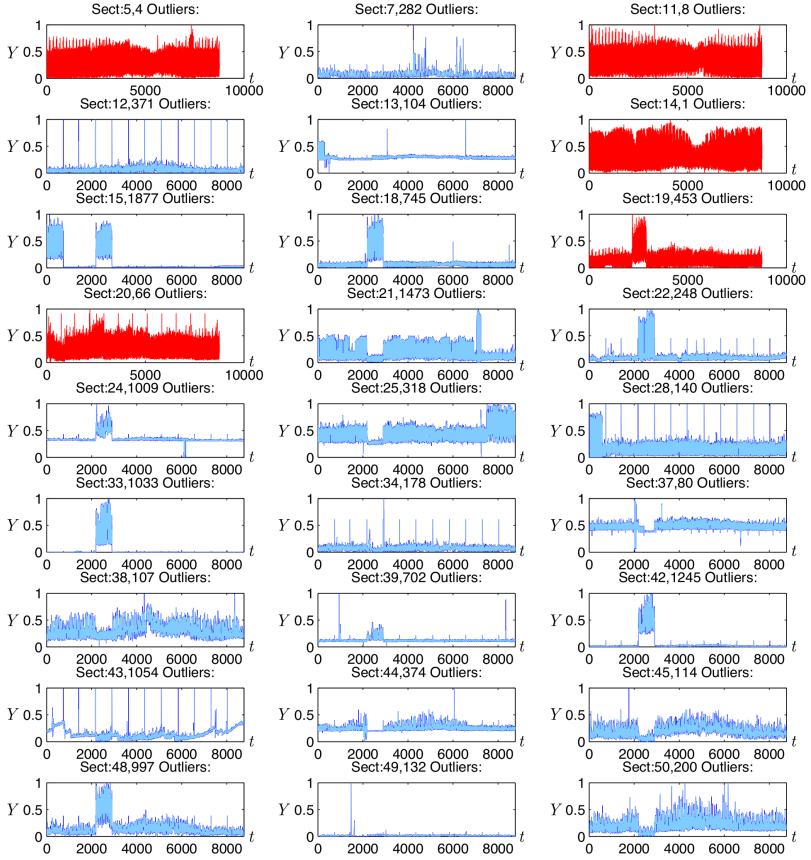


Figure 24: Raw time series generated by different flowmeters during the year 2012.

The characteristic trend of the selected time series is shown in the plot of Figure 26 and 27.

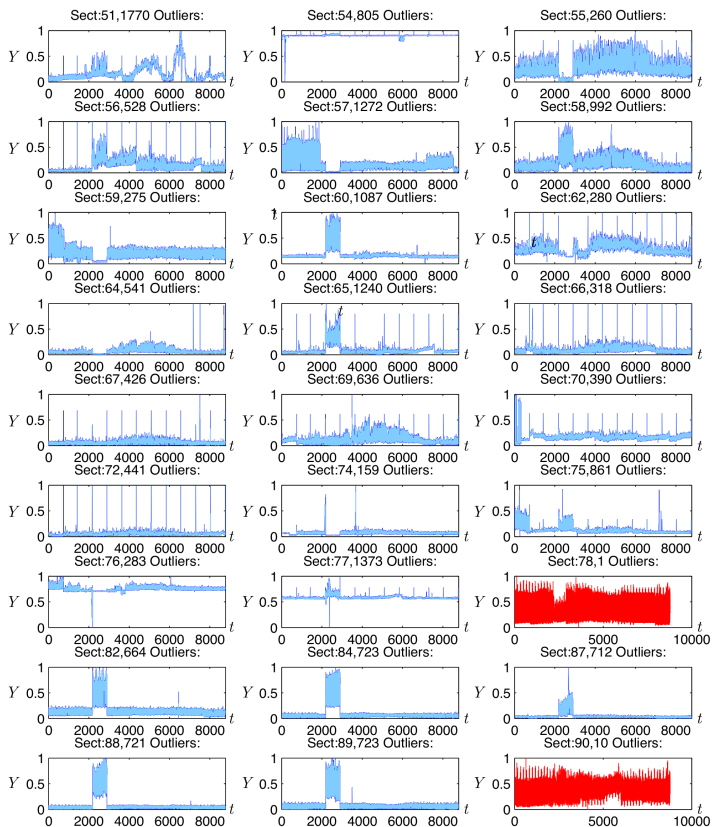


Figure 25: Raw time series generated by different flowmeters during the year 2012

The fitted green plain line describes the trend of each series with the slope and intercept components proper of the linear regression. The values of the series of interest are presented in Table 3.

A light trend near to zero is noticed in the series, nevertheless it is not possible to assume constant trend in the long term as soon as we do not have more information about the next years. The trend of the Sector 19

Sector	Outliers
5	4
11	8
14	1
19	453
20	66
78	1
90	10

Table 2: Number of outliers detected by the modified Thompson *tau* technique with a significance of $\alpha = 0.01$

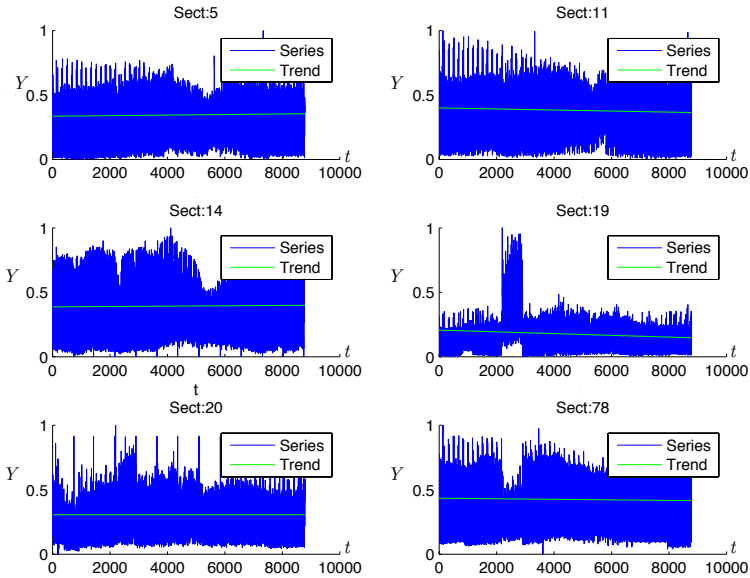


Figure 26: Linear regression for illustrating the trend of the selected time series

is severely affected for an irregular period that makes to have a possibly false sense of the trend.

Regarding the seasonality, Figures 28 and 29 show the strongest auto-correlation at lag 24 for all the time series, which confirms the water de-

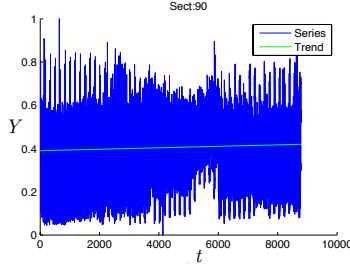


Figure 27: Linear regression for illustrating the trend of the selected time series

Sector	Slope (1×10^{-5})	Intercept (Mean)
5	0.2214	0.3351
11	-0.4062	0.3998
14	0.1380	0.2066
19	-0.6750	0.3054
20	0.008	0.4340
78	-0.2145	0.3900
90	0.3389	0.3900

Table 3: Trend described by the slope and intercept components of linear regression

mand cycles consumption are repeated in daily basis. Once the database is described, the next Subsection presents the results of the different algorithms developed.

5.2 Experiments

This subsection presents the performance presented by the different Multi-Model Predictor algorithms: the quantitative and qualitative Multi-Model predictors (**QMMP+NF**, **QMMP+kNN**) and the ANN Multi-Model Predictor (**RBFMMP+OR**). These algorithms are tested and compared with the selected time series described previously.

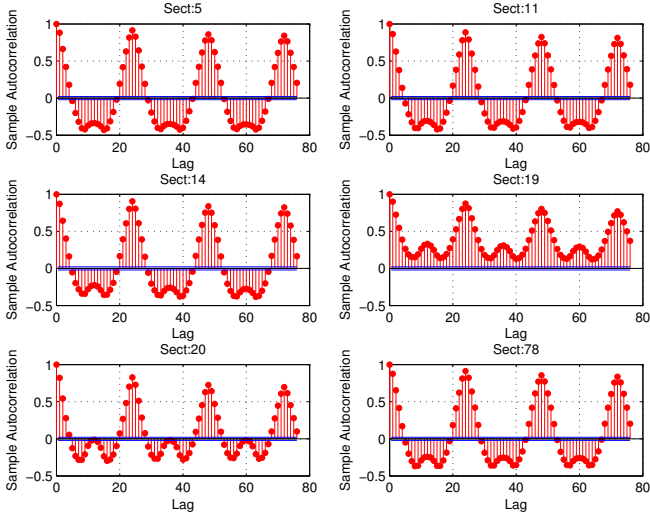


Figure 28: Autocorrelation plots of different sectors

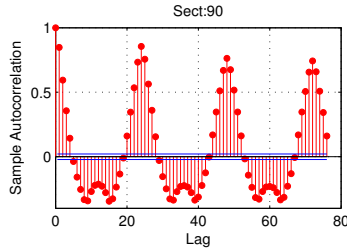


Figure 29: Autocorrelation plot of sector p10025

5.2.1 Validation and Performance Comparison of QMMP Algorithms

The performance of **QMMP+kNN** and **QMMP+NF** depends on the selected prediction strategy. As we mentioned before, a SARIMA pattern is selected to produce the quantitative forecast and an activity calendar or a kNN to produce the qualitative forecasting. To select the qualita-

tive prediction strategy, a two fold cross validation is performed over a the Training-Validation set from the full mode sequence. This analysis helps to understand the dynamics that follows the mode sequences. The organisation of the sets are described in Figure 30.

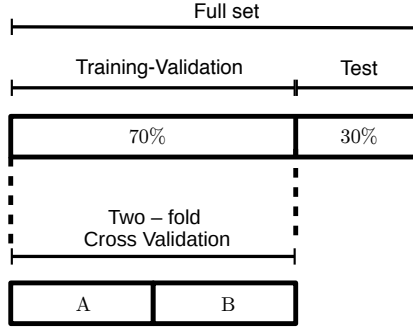


Figure 30: Organisation of the training-validation and test sets

The training-validation is divided in datasets A and B taken from the 70% of the total data. For this task the time series \mathbf{Y}_{full} , is decomposed in its qualitative and qualitative components \mathbf{Z} and \mathbf{X} by using the Equation 4.12 and 4.13 respectively. The parameter τ is set to 24 according to the seasonality observed in the correlation plots of Figure 28 and 29, the qualitative time series \mathbf{X} is classified using k-means. The number of classes is selected according to the maximum silhouette coefficient criteria. Different number of classes are tested with $K_{test} = \{1, \dots, 10\}$. Their silhouette values are plot in the Figure 31, where all the time series considered matches with a number of classes $k = 2$.

The centroids of each class obtained by k-means represent the different nominal consumption patterns and are used as unitary mode prototypes. The classification labels \mathbf{C} are represented in Figure 32, where each monochromatic square represents a pattern mode label.

We consider the calendar \hat{F}_{cal} (Equation (4.21)) and kNN qualitative mode predictor (Section 3.4.2) to be chosen. The mode strategy that de-

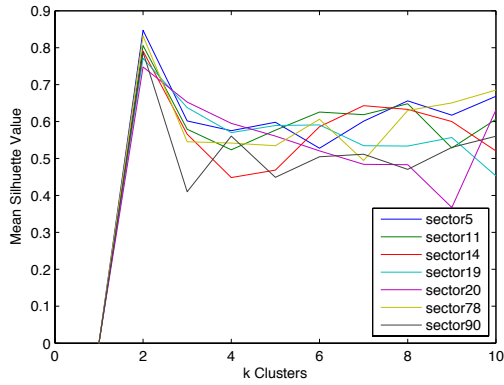


Figure 31: Mean silhouette coefficient values for different values of k in k -means and different time series of water demand.

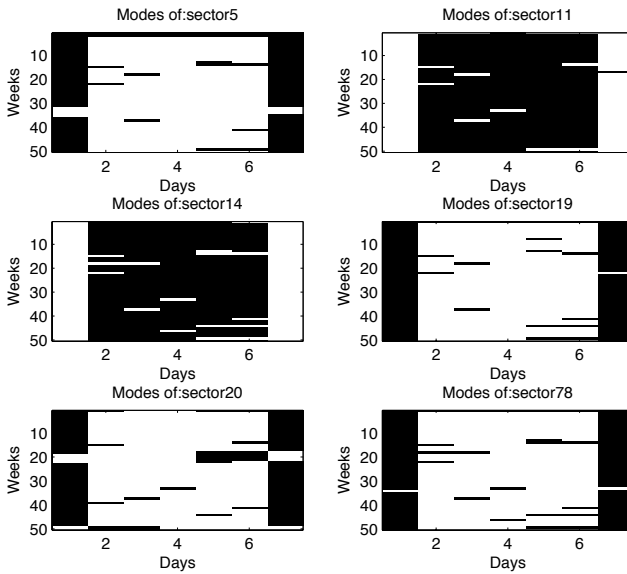


Figure 32: The two pattern modes of different sectors. The week starts on Sunday.

scribes and predicts better the mode sequences is selected.

The cross-validation estimates the predictability of the pattern mode of both strategies. The strategy is selected according to the minimisation of the average pattern prediction errors criteria. The two iteration cross-validation is performed in this way: the first iteration considers the subset B as the training set and the set B is the validation set, and for the second iteration, B is used as training set and B as the validation set. After finishing the two iterations, the errors obtained are averaged. The results are shown in the Table 4.

Sectors	Iteration 1		Iteration 2		Mean		Suggestion
	kNN	Calendar	kNN	Calendar	kNN	Calendar	
5	5	5	13	0	9	2.5	Calendar
11	20	4	15	7	17.5	5.5	Calendar
14	5	5	11	17	8	11	kNN
19	3	5	14	1	8.5	3	Calendar
20	4	1	15	2	9.5	1.5	Calendar
78	4	3	7	2	5.5	2.5	Calendar
90	7	5	18	2	12.5	3.5	Calendar

Table 4: Two iteration cross validation for kNN and Calendar mode estimation

In both iterations the parameters m and ε defining the length and neighbourhood of the qualitative delay vectors of kNN are optimised using the Equation 5.1

$$\arg \min_{m, \varepsilon} \sum_{T=1}^N dist(\hat{\mathbf{C}}_{T+H}, \mathbf{C}_{T+H}) \quad (5.1)$$

setting the prediction horizon one day ahead $H = 1$ (prediction of one pattern mode ahead), where $dist$ is the Hamming distance taking into account that \mathbf{C} is a sequence of labels and the prediction is performed by using the Equation 3.68. Table 5 includes the parameters that optimise the prediction with kNN for each iteration. The distance of the neighbourhood and size of the delay vector for kNN mode estimator are those that minimises the objective function expressed by (4.18) subject to $\varepsilon = \{0.01, 0.02, \dots, 0.1\}$ and $m = \{2, \dots, 10\}$.

	Iter 1	Iter 2
Sector	m	m
14	10	5
90	8	6
78	8	9
20	10	9
11	6	8
5	10	9
19	8	9

Table 5: m parameters that optimize the mode prediction with kNN. $\varepsilon = 0.01$ was chosen for all time series

This test suggests the use of the calendar for all time series, with the exception of the Sector 14 that is more suitable to be predicted using kNN.

Performance of QMMP + kNN and QMMP+NF

For the algorithm **QMMP + kNN**, the data set is also divided in two data sets. The training and testing set with the 70% and 30% from the total data respectively. The preparation phase is performed with the training set by means of the decomposition, pattern mode estimation, SARIMA and kNN parameter tuning.

In the case of the algorithm **QMMP+NF**, the data set is divided in 3 sets. The preparation phase of is similar to the **QMMP + kNN**. With the difference that the half of the testing set is used for tuning the noise filter. The organisation of the different data sets can also be observed in Figure 33.

The parameters m , ε and SARIMA structures of the two algorithms are shown in Table 6 that consider the search space for the kNN qualitative forecaster is defined by $m = \{1, \dots, 20\}$ and $\varepsilon = \{0.01, 0.02, \dots, 1.00\}$. For the noise filter parameters, the search space is defined by $\alpha = \{0, 0.1, 0.2, \dots, 1\}$ and $\kappa = \{1, 2, \dots, 40\}$. All those values are tested using the validation set and selected those that minimise the mean of all the inde-

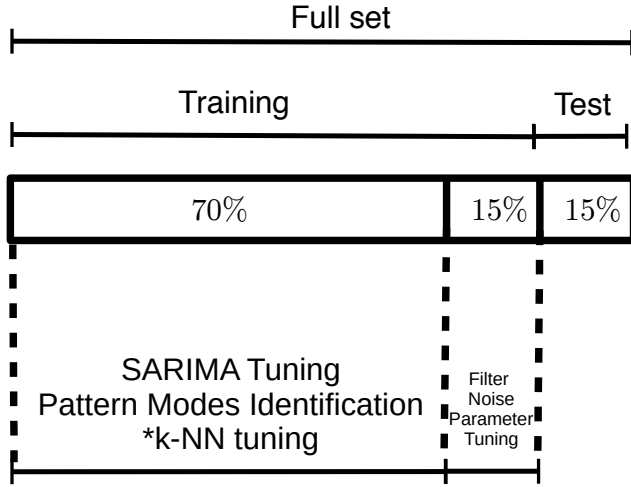


Figure 33: Training and testing data organisation

pendent $RMSE_h$ described by Equation 5.2.

$$\arg \min_{\alpha, \kappa} \overline{RMSE_h} \quad (5.2)$$

The two algorithms are compared with well known forecasting models, Double Seasonal Holt-Winters (DSHW) and RBF Neural Networks (RBF-ANN).

For the Holt-Winters, two parameters associated with two periods are fixed. Since Holt-Winters manages hourly data, period τ_1 and period τ_2 parameters are set to 24 and 168 for the daily and weekly period. For the RBF-ANN, different structures are implemented with 192 neurons in the hidden layer, fixed widths with $\sigma = 1$, and input layer of size 24 and the output layer of size 24 ($h = 24$).

Finally to measure the performance of the algorithms is used the \overline{RMSE}_{24} (Equation 3.79). The \overline{RMSE}_h produced by the **QMMP**, **QMMP+NF**, **QMMP+kNN**, **RBF-ANN** and **DSHW** using the test set for $h = 24$

steps ahead, α and κ best values for the **QMMP+NF** are reported in Table 7.

Sectors	SARIMA Models						kNN	Noise filter	
	p	d	q	P	D	Q	m	α	κ
5	[2,4]	1	[1]	[0]	0	[7]	9	0.85	70
11	[2,4]	1	[1]	[0]	0	[7]	14	0.65	70
19	[2,4]	1	[1]	[0]	0	[7]	13	0.55	42
20	[2,4]	1	[1]	[0]	0	[7]	13	0.8	70
78	[1,2,3,19]	1	[0]	[0]	0	[7]	10	0.6	70
90	[2,4]	1	[1]	[0]	0	[7]	8	0.7	70
14	[0]	1	[2, 7]	[0]	0	[7]	7	0.6	70

Table 6: Parameters for the Noise Filter, kNN Qualitative Forecaster and SARIMA models for each time series found.

Sec	QMMP	QMMP+NF	QMMP+kNN	RBF-ANN	DSHW
5	0.0436	0.04	0.0494	0.0563	0.1009
11	0.0492	0.047	0.051749	0.0649	0.0755
19	0.0248	0.0239	0.029168	0.0333	0.0313
20	0.0494	0.0473	0.050302	0.0578	0.0555
78	0.0404	0.0359	0.041641	0.0572	0.0512
90	0.0632	0.0469	0.0640	0.0627	0.0609
14	0.0723	0.0727	0.0418	0.0558	0.0592

Table 7: \overline{RMSE}_{24} performance comparison of the different algorithms

In order to compare graphically the performance of the **QMMP+NF**, Figures 34 - 47 illustrate the improvement that the filter produces: the first column of plots shows the prediction errors $RMSE_{24}$ at each time t in hours along the testing set comparing the performance of the Noise Filter (**QMMP+NF**) (red line) with the original Qualitative and Quantitative forecasting (**QMMP**) (Black dotted line) taking into account the pattern mode calendar. For the second column, a prediction sample 48 hours ahead is plot comparing the actual (black line), with the normal (**QMMP**) (blue dotted line) and filtered predictions (**QMMP+NF**) (red

line). It is observed that the noise filter reduces in general the prediction error since the red line is the most of the time lower than the black line representing the prediction with SARIMA pattern without filtering.

Discussion

The proposed architecture can be generalised for those time series that present clear patterns in regular time lapses and can be validated as in Section 6.1. The SARIMA pattern is a simple and effective method that fit in the proposed Multi-Model **MMP** architecture. It works quite good for the short term water demand forecasting. The general architecture of this methodology is very flexible, although we explored some configurations. However more work exploring suitable algorithms for decomposing the time series in other simpler forms and predicting them should be done. The architecture allows to substitute and plug different forecasting models for the qualitative and quantitative components. For example, the quantitative series is possible to be modelled with **DSHW**, **ANN** among others. Also the qualitative component is not limited to be modelled either with the calendar or **kNN**. As an example, the structure organisation of the multi-model allows to implement straightforward a mix of the **MMP+kNN** and **MMP+NF** approaches for increasing the prediction accuracy.

Conclusions

This subsection has shown a validation strategy based on the two iteration cross-validation for choosing a qualitative mode predictor for proposed Multi-Model Predictor based on the decomposition of qualitative and quantitative components. Also two algorithms are compared against **RBF-ANN** and **Holt-Winters**. **kNN** mode predictor is shown as an alternative of pattern prediction when the calendar based pattern prediction performs poor. The algorithms and models can be chosen a priory observing the performance presented in the testing set. A filter that distinguishes the noise from the structure of the past residuals to improve the prediction accuracy is introduced. It is demonstrated in practice that dis-

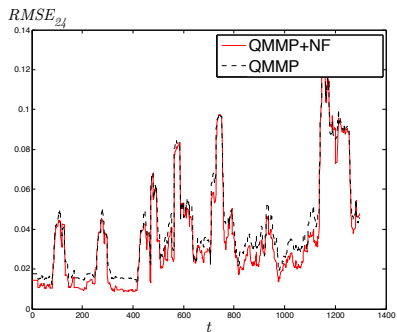


Figure 34: Error along the testing set of Sector 5

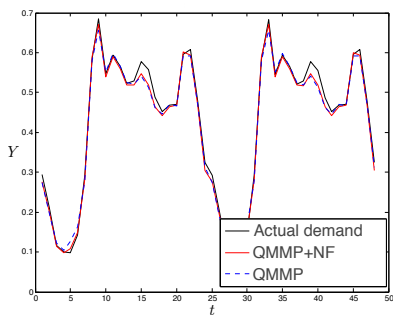


Figure 35: Prediction sample of 2 days ahead of Sector 5

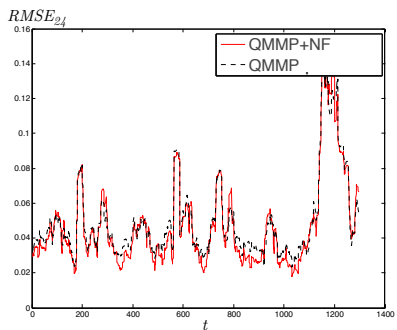


Figure 36: Error along the testing set of Sector 11

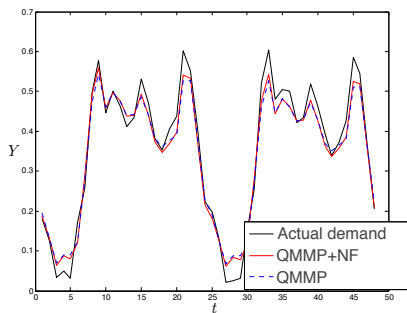


Figure 37: Prediction sample of 2 days ahead of Sector 11

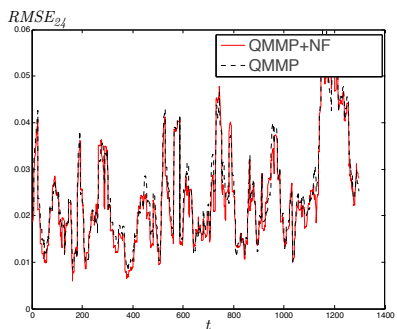


Figure 38: Error along the testing set of Sector 19

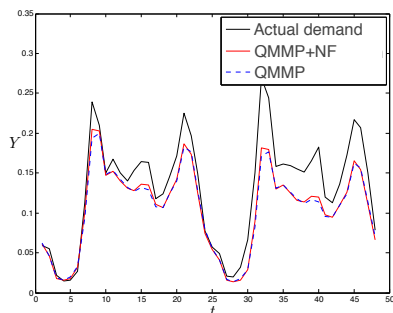


Figure 39: Prediction sample of 2 days ahead of Sector 19

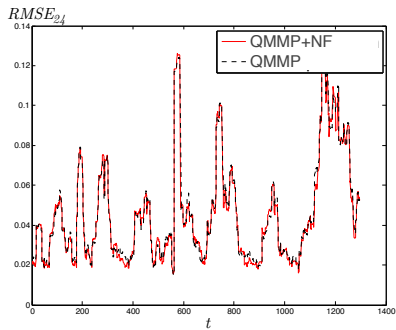


Figure 40: Error along the testing set of Sector 20

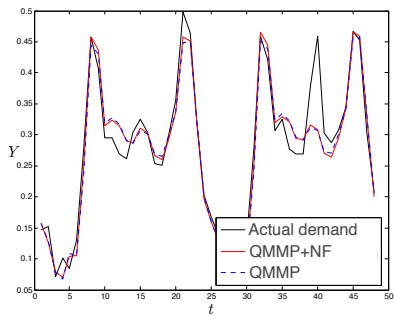


Figure 41: Prediction sample of 2 days ahead of Sector 20

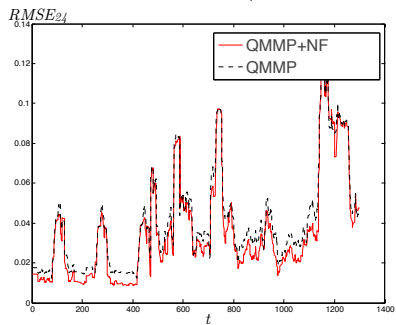


Figure 42: Error along the testing set of Sector 78

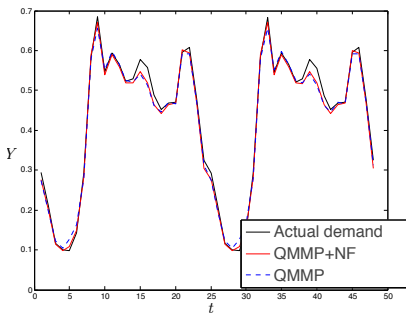


Figure 43: Prediction sample of 2 days ahead of Sector 78

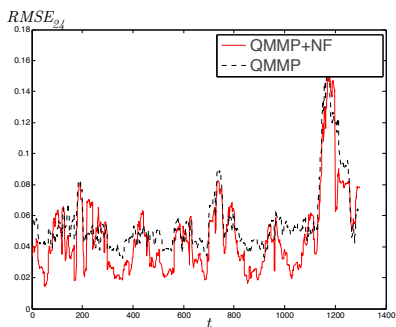


Figure 44: Error along the testing set of Sector 90

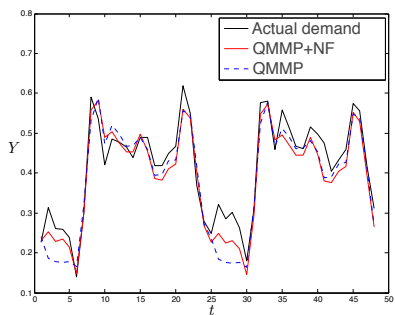


Figure 45: Prediction sample of 2 days ahead of Sector 90

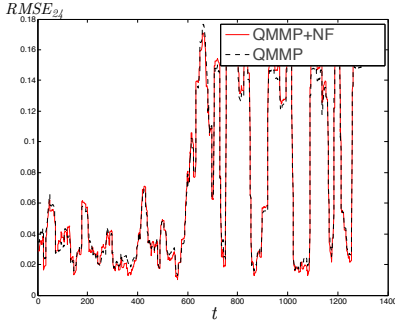


Figure 46: Error along the testing set of Sector 14

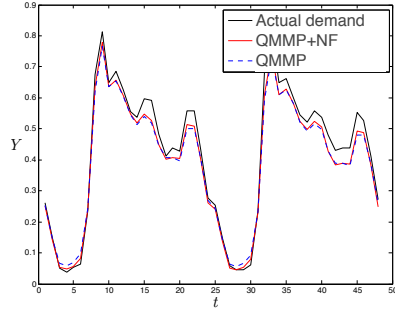


Figure 47: Error along the testing set of Sector 14

covering structure from the past errors may improve the performance in general. The implementation of the filter processing may produce more flexible and adaptable forecasting models that fit the gradual appearance of unobserved and unmodelled dynamics that may occur in real systems.

As future work the authors suggest to explore the performance of the kNN mode predictor plus the noise filter for improving the pattern prediction. Also is suggested the exploration of the other kind of forecasters and filters and the study.

5.2.2 Validation and Performance of the Multi-Model Forecasting Using RBF Artificial Neural Networks with an On-line Mode Recognition

In this subsection, another Multi-Model Predictor implementation, the **RBFMMP+OR**, is applied to the water demand time series described previously in Subsection 5.1.

The **RBFMMP+OR** is implemented with the algorithms described in the Section 4.2 setting the parameters as follows: for the **Split Time Series** and **Mode Recognition** modules, the split point is set to $\tau = 12$ and the mode recognition implements the nearest neighbour rule. For the **Feature Extraction** module implements the discrete derivative or differ-

entiation of the time series. The **Pattern Classifier** implements k-Means. The number of clusters k is defined by the maximisation of the average silhouette coefficient. For the **Multi-Model** module predictor, the RBF-ANN structure has input layer of size $k + 24$, the hidden layer has 192 neurons with width $\sigma = 1$ and an output layer size 24 where each output neuron is the prediction for each time step in the next 24 hours.

The data is divided in two sets; 70% for training and tuning the different parameters of the models, and the 30% for validation. The Figure 48 shows the average silhouette values obtained for different sectors and different number of clusters k . The **Classifier** module uses k-Means algorithm.

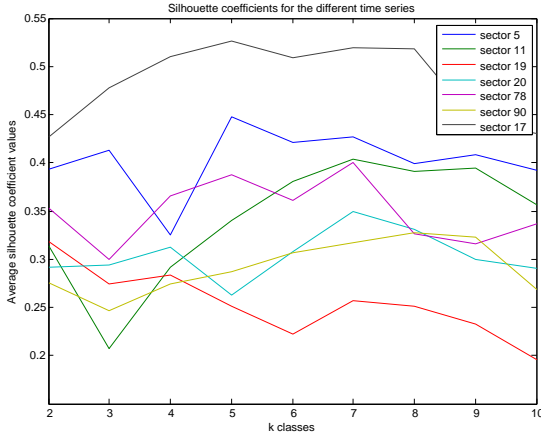


Figure 48: Mean silhouette coefficient values for different values of k in k-means and different time series of water demand.

The results including the number of clusters selected are indicated in Table 48 where the \overline{RMSE}_h given by the Equation 3.79 is used for measuring the accuracy of the forecasting in the validation set. The proposed approach **RBFMMP+OR** is compared with:

- A Seasonal ARIMA Pattern (SARIMA) (**QMMP**) based on the work of (QPG⁺10) with a SARIMA model structure is defined previously

in the Table 6.

- A Double Seasonal Holt-Winters (DSHW) (Kal04) with period τ_1 and period τ_2 set to 24 and 168 for the daily and weekly period, respectively.
- A standard RBF Artificial Neural Network **RBF-ANN** considering $m + k$ inputs and h outputs, with $m = 24$.

The Table 8 presents the results of this comparison. The columns of the **RBFMMP+OR** contains the modes identified from the Figure 48.

Sect.	QMMP	DSHW	k modes	RBFMMP + OR	RBF-ANN 24 + k inputs h outputs
5	0.0453	0.1009168	5	0.0522	0.0526
11	0.051719	0.07225436*	7	0.0658	0.0606
19	0.027823	0.03130634	2	0.0336	0.037
20	0.045474	0.05559293	7	0.0543	0.0557
78	0.040847	0.0512956	7	0.0546	0.0527
90	0.059176	0.0609	8	0.0624	0.0599
14	0.0651	0.05646	6	0.0433	0.0513
Avg.	0.0479	0.0612	6	0.0523	0.0528

Table 8: \overline{RMSE}_{24} indicator for each time series produced by **QMMP**, **DSHW**, **RBFMMP+OR** and **RBF-ANN**

Evaluating the average performance of each algorithm for all the series, the Table shows that **QMMP** is the algorithm with the best average performance with an average error of 0.0479. The second best is the proposed algorithm **RBFMMP+OR** with an average performance of 0.0523, the third best algorithm is the **RBF-ANN** with an average performance of 0.0528. The **DSHW** is the algorithm with the worst performance with an average performance of 0.0612. Comparing with percentages, **QMMP**, **RBFMMP+OR**, and **RBF-ANN** are %21, %13.7 and %14.5 better than **DSHW** performance. The asterisk written in the performance box of the **DSHW** with regard to Sector 11 means that it displays an unstable behavior for a set of values of the time series. So, in order to have a consistent

error measurement, the unstable section was omitted. This kind of situations reinforces the need of exploring filtering and reconstructing the time series. Figures 49-55 present the prediction samples produced by the **RBFMMP+OR** 24 hours ahead during the day 298. The red dotted line is the prediction and the solid black line is the real water demand during that period.

Another experiment is presented analysing the performance of **RBF MMP + OR** at different prediction horizon values h with the same algorithms presented in Table 8. **RBFMMP + OR** shows better performance than the traditional **RBF-ANN**, specially for the series of Sector 14. The experiment consists in testing different **RBF MMP +OR** and **RBF-ANN** architectures, such that the structures having an input layer of size 24 plus the mode input size k , 192 neurons in the hidden layer, and variable outputs size considering the desired horizons $h = \{1, 2, 3, 4, 5, 6, 12, 24\}$. The mean silhouette values for each prediction horizon h is shown in the Figure 56 and the results including the selection of k are presented in Table 9.

Horizon h	QMMP	DSHW	k	RBFMMP+OR k-Means 24 + k inputs h outputs	RBF -ANN 24 + k inputs h outputs
1	0.0422	0.04513	5	0.0173	0.0240
2	0.0456	0.04729	5	0.0217	0.0270
3	0.0483	0.04885	5	0.0244	0.0307
4	0.0506	0.05016	5	0.0260	0.0333
5	0.0528	0.05131	5	0.0283	0.0350
6	0.0551	0.05234	5	0.0283	0.0366
12	0.0651	0.05646	6	0.0345	0.0425
24	0.0587	0.05922	6	0.0433	0.0511

Table 9: \overline{RMSE}_h indicator for each method for the hourly time series of the Sector 14.

Table 9, shows that **RBFMMP+OR** is better than **RBF-ANN** at different prediction horizons. **QMMP** and **DSHW** performs quite similar but not better than the propped framework. The good accuracy at shorter

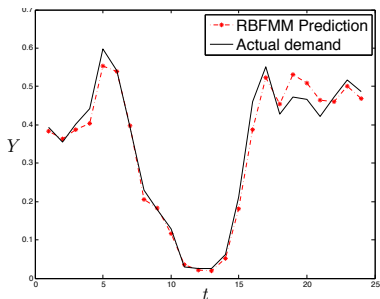


Figure 49: Prediction sample Sector 5

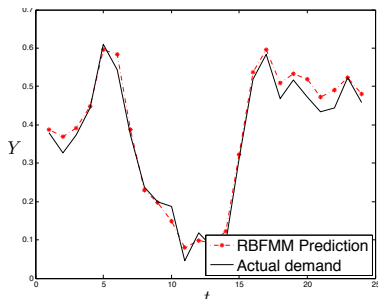


Figure 50: Prediction sample Sector 11

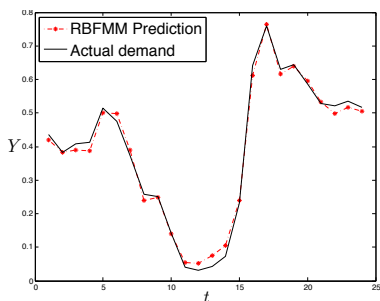


Figure 51: Prediction sample Sector 17

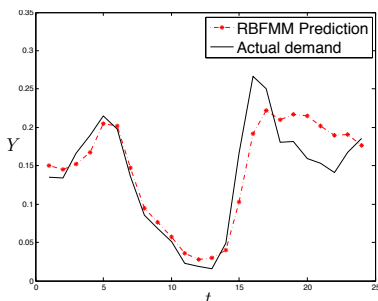


Figure 52: Prediction sample Sector 19

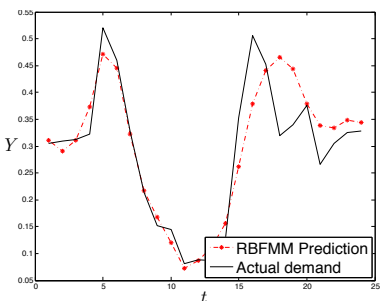


Figure 53: Prediction sample Sector 20

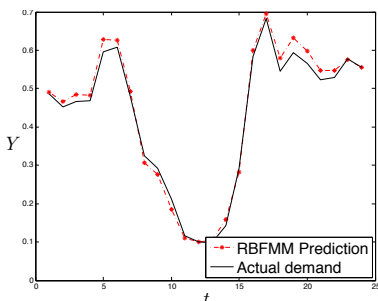


Figure 54: Prediction sample Sector 78

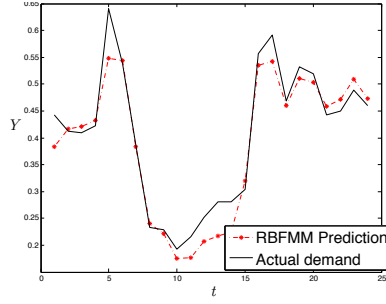


Figure 55: Prediction sample Sector 90

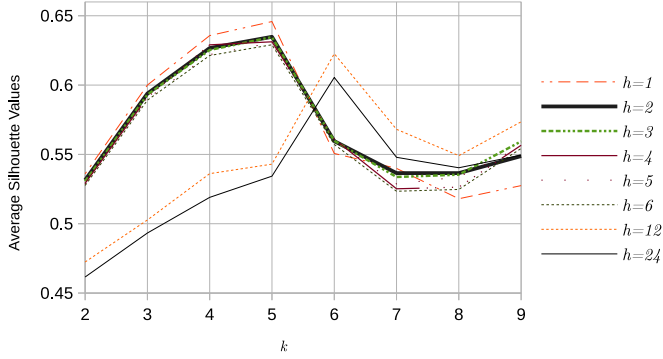


Figure 56: Mean silhouette value for varying values of k and variable prediction horizon h

prediction horizons, is because special architectures are trained to predict at specific horizon avoiding the accumulation of forecasting errors, unlike forecasting models such as DSHW or ARIMA that are not able to produce a forecast horizon longer than one without feedback.

5.2.3 Discussion

This approach basically proposes a methodology to train and implement a Multi-Model Predictor with classified temporal data using RBF-ANN

for time series with defined periodicity. This approach is based on the concept of regimes or operation forecasting modes and required mode recognition mechanism to identify the current mode on-line. One of the main drawbacks is the selection of a good value for the split point. A short split point value leads to have short prototypes and short overlaps between consecutive segments producing a big number of classes. A big split point leads to have long prototypes maybe difficult to recognise when the number of observations is short producing poor mode estimations and consequently poor predictions. Remember that the split point defines the minimum amount of observations stored in the queue for estimating the current mode. Although it is not the best model for minimising the error for all the time series studied in this thesis, it performs better than the traditional RBF-ANN. The author believes that the use of more sophisticated algorithms for clustering, feature extraction and prediction might improve the performance even more. As future work, other implementations should be studied extending the experimentation with more algorithms such as feature extraction and, dimensionality reduction algorithms. We can conclude the model is suitable for stationary time series presenting defined seasonal patterns along the time.

Chapter 6

Conclusions

The main contribution of this thesis is the introduction of general Multi-Model Predictor (**MMP**) architecture. Different implementations were proposed following the qualitative and quantitative decomposition of the data and a Multi-Model forecasting based on Machine Learning methodologies with the implementation of an On-Line Mode Recognition.

6.1 General Conclusions

Time series modelling is related to the black-box modelling of dynamical systems when the model that produces such dynamics is completely unknown. For addressing this problem, different tools from statistic and computer science are used nowadays to describe the data sequence using general models.

In statistics, classical methods as the parametric models like moving average, Auto-Regressive, exponential smoothing models that assumes certain regularity in the dynamic composition of the data can be found, in computer science nonparametric models that do not need to satisfy any assumption about used data as Support Vector Machines, multilayer perceptron and Artificial Neural Networks which due to their characteristics are used to approximate any function.

Time series extracted from the nature might present complex behav-

ior difficult to be modelled just with one single linear model. In order to obtain a more accurate approximation, it is assumed that the behavior can be approximated by interpolating several Linear Time Invariant (LTI) models, such that each one is used to model scenarios or regimes locally of the dynamical system. This fact is related to the multiple modelling approach of nonlinear dynamical systems.

Based on this idea, this thesis addressed the Multiple-Modelling problem by defining a general framework which may adopts a collection of statistical and machine learning algorithms for producing as accurate predictions as possible of the system by identifying and modelling the local regimes.

The work of this thesis has some advantages over the proposed methodologies mentioned in Chapter 2. The proposed multi-modelling approach does not depend completely of the structured periodicity of the data, overall when the global model (presented in the sequence of modes or regimes) is based on nonparametric forecasting methodologies which learns from the past information to predict the next mode. These algorithms are based on the knowledge acquired by learning from the historical behavior of the data from predefined activity modes governed by a calendar. In other words, these prediction algorithms are based on the *common sense* that come from the predefined calendar of the human activity, on the analysis of the past information to learn and on predicting the behavior of the next time step as shown with the qualitative kNN predictor.

6.2 Particular Conclusions

The particular conclusions related to the algorithms developed in this thesis are:

- **Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition using SARIMA and kNN (MMP+kNN):** This algorithm presents better performance in the water demand time series of a sector that does not follow the dynamic of the prede-

defined activity calendar at all. The implementation outperforms the SARIMA pattern algorithm, RBF-ANN, and Holt-Winters for this case.

- **Multi-Model Predictor Based on the Qualitative and Quantitative Decomposition of the Time Series including a Nonlinear Filter (MMP+NF):** This Multi-Model implementation includes a nonlinear filter. The nonlinear filter is used with water demand time series which qualitative behavior approximates the human calendar activity of working and resting days. The nonlinear filter was able to find new nonlinear structure of the qualitative behavior of the time series. The integration of this mechanism provides of interesting advantages over other algorithms in real situations where the dynamics of the system is changing gradually along the time, such that with traditional approaches, the new data should be analysed in order to decide if the current model is still valid or not. As soon as is detected a dynamic change, it might be required to construct a new forecasting model. The proposed model is able to detect the mismatches to correct the next prediction pattern. This algorithm is more robust to the model mismatches in the sense that presents adaptability as soon as the new information provided by the pattern mismatches is collected and processed.
- **Multi-Model Forecasting Using Radial Basis Function Neural Networks with an On-Line Mode Recognition (RBFANN+OR):** This implementation proposes at the same time a general framework for constructing a Multi-Model Predictor based completely in machine learning algorithms. The disadvantage of this approach is that sometimes produces false positives regarding the forecasting mode.

The three methodologies based on the proposed general Multi-Model Predictor framework has some disadvantage that should be solved. The mode prediction may produce false positives. This is worsen when the assumption of the certainty of the data is not correct at all. For example, the prediction fails when the strategy is to trust in the pattern calendar

activity when in reality the prediction mode is governed by other kind of rules. In this situation the prediction will not be robust enough. In order to solve this problem, it is possible to be selective by associating certain algorithms or prediction schemes to specific time series. This can be performed off-line executing a cross validation methodology as we did in Section 5.1. If the prediction scheme it is required to be adaptive taking decisions in an on-line manner. Moreover, methodologies for exploiting in real time the information collected so far should be considered for the prediction of the next mode for the global modelling.

6.3 Future Work

As future work the author proposes to follow the research line of global modelling based on time series models including modes that can be implemented in the Multi-Model Predictor architecture. The prediction accuracy of the next mode depends on the knowledge assumption of the data. It might be a stochastic Markov process, deterministic rule based prediction, as previously mentioned in Chapter 3. A correct general assumption of the sequential data provides robustness to the prediction. The author believes that the integration of different (possibly correlated) information sources as temperature, humidity and pollution levels as auxiliary information for the qualitative forecasters can be integrated and exploited for improving the global mode prediction accuracy. Bayes based tools such as Bayesian Networks allow the integration of all these diverse information because they are able to capture the probabilistic dependency of random variables given a set of observations (HKP11). These kind of models are capable to predict the operation modes using the Bayesian inference; in other words, it is possible to estimate the posterior probability of unknown variables given a set of observations from diverse sources considered as random variables. These variables might be associated with the operation modes or regimes.

Another research direction, is the generalisation of the qualitative and quantitative method decomposition (Section 4.3) generating several qualitative and quantitative time series associated with specific subpe-

riod of water demand pattern.

The proposed decomposition is useful when certain segments (or sub-periods) of a period, obeys to different behavior. For example, in the work of (BVVS13), M. Bakker et al. detected in the central and Southern part of Netherlands, an increment of the normal water demand from late afternoon to evening in summer days. This increment observed in these specific hours of the day is called, variant of season water demand pattern. This pattern is produced by people sprinkling their gardens. Therefore, the proposed forecasting method works differently for this part of the day.

Therefore the proposal is to extend the decomposition algorithm splitting the size period τ in subperiods of size τ' . The selected subperiod size τ' is selected satisfying

$$\{\exists \tau', \text{mod}(\tau, \tau') = 0 \text{ and } \tau' \leq \tau\} \quad (6.1)$$

where if τ is a multiple of τ' then $\text{mod}(\tau, \tau') = 0$ and then, τ'/τ number of qualitative and quantitative different time series are obtained. For example: if $\tau = 24$ assuming the time series shows seasonality each 24 hours and $\tau' = \tau/2$. Then, it is possible to obtain a collection of two qualitative and two quantitative time series associated with specific period segments $\{1, \dots, \tau'\}$ and $\{\tau' + 1, \dots, \tau\}$. The generalisation of this kind of decomposition (shown in Equation 4.12 and 4.13) is defined by Equations 6.2 and 6.3:

$$X_T^a = \sum_{i=(T-1)\tau+(a-1)\tau'+1}^{(T-1)\tau+a\tau'} Y_i \quad (6.2)$$

$$Z_T^a = \frac{\{Y\}_{i=(T-1)\tau+(a-1)\tau'+1}^{(T-1)\tau+a\tau'}}{X_T^a} \quad (6.3)$$

where $a = \{1, \dots, \frac{\tau}{\tau'}\}$ indicates the subsection part of size τ' , and T the T -th subsection of kind (or mode) a . Such transformation produces $\frac{2\tau}{\tau'}$ simpler time series, of which a number of $\frac{\tau}{\tau'}$ are quantitative and same number are qualitative with subpatterns of size τ' . The sequence of el-

ements associated with each subsection a for the qualitative time series are set in \mathbf{X}^a expressed in Equation 6.4.

$$\mathbf{X}^a = \{X_1^a, \dots, X_T^a, \dots, X_{\frac{n}{\tau}}^a\} \quad (6.4)$$

The set of qualitative time series is set to \mathbf{Z}^a expressed by Equation 6.5

$$\mathbf{Z}^a = \{Z_1^a, \dots, Z_T^a, \dots, Z_{\frac{n}{\tau}}^a\} \quad (6.5)$$

where Z_1^a, X_1^a are the first element, Z_T^a, X_T^a the current segments at time T and $Z_{\frac{n}{\tau}}^a, X_{\frac{n}{\tau}}^a$ the last element of the series. For each qualitative \mathbf{X}^a and quantitative \mathbf{Z}^a time series a local forecasting model is built for each sequence, f_a^{ql} and f_a^{qn} respectively are part of bigger models \mathbf{F}^{ql} and \mathbf{F}^{qn} described by Equation 6.6 and 6.7.

$$\mathbf{F}^{ql}(\mathbf{X}_T) = \begin{cases} f_1^{ql}(X_T^1, \dots, X_{T-\alpha}^1) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = 1 \\ f_2^{ql}(X_T^2, \dots, X_{T-\alpha}^2) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = 2 \\ \vdots & \\ f_a^{ql}(X_T^a, \dots, X_{T-\alpha}^a) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = a \end{cases} \quad (6.6)$$

$$\mathbf{F}^{qn}(\mathbf{Z}_T) = \begin{cases} f_1^{qn}(Z_T^1, \dots, Z_{T-\alpha}^1) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = 1 \\ f_2^{qn}(Z_T^2, \dots, Z_{T-\alpha}^2) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = 2 \\ \vdots & \\ f_a^{qn}(Z_T^a, \dots, Z_{T-\alpha}^a) & \text{if } \text{mod}(\lfloor \frac{\tau'(t+1)}{\tau} \rfloor, \frac{\tau}{\tau'}) = a \end{cases} \quad (6.7)$$

where α limits the previous days taken into account for producing the prediction. These models belong to a Multi-Model Predictor which the composition of both produces a short term forecast of τ' steps ahead associated with each segment a . The prediction produced is described by Equation 6.8.

$$\mathbf{F}^{qn}(\mathbf{Z}_d) * \mathbf{F}^{ql}(\mathbf{X}_T) = \hat{Y}_{T\tau+(a-1)\tau'+1}, \dots, \hat{Y}_{T\tau+a\tau'} \quad (6.8)$$

where $\mathbf{F}^{qn}(\mathbf{Z}_T) * \mathbf{F}^{ql}(\mathbf{X}_T)$ is the composition of the qualitative and quantitative predictions.

Figure 57 shows an example where $\tau = 24$ and $\tau' = \tau/4$ producing 4 subsegments of size τ' composing the full cycle. The past data is limited setting $\hat{\alpha} = 3$ representing the days (or cycles) considered (or cycles) in the past. The segments $a = \{1, 2, 3, 4\}$, works as model selector associated with each time series and forecasters introduced in Equations 6.1 - 6.7 to the generalised multi-model decomposition. The dotted lines are the respective forecasts produced by generalised the Multi-Model Predictor (Equation 6.8) based on the qualitative and quantitative decomposition of the series.

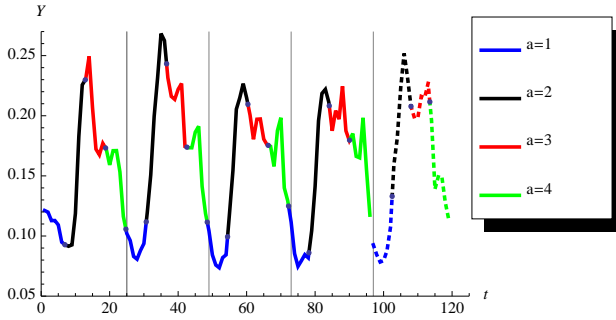


Figure 57: Segments of the generalised decomposition

Thus, from these results it can be concluded that the framework proposed still requires a lot of experimentation to do in order to the proposed approaches can be applied to other purposes as for example, the modelling of the electricity demand.

Appendix A

List of Symbols

a	Model parameter selector
a, b	Objects
A, B	Sets of objects
\mathbf{A}	Set of model parameter selectors
b	Trend component of Holt-Winters
C	Label element
\hat{C}	Estimated label element
\mathcal{C}	Cluster
C_t	Multiplicative coefficient
\mathbf{C}	Vector of labels
$\mathbf{C}^I, \mathbf{C}^O$	Number of runs of labels O, I
$dist$	Distance function
\mathbf{D}_t	Delay vector
f	Function
f^{ql}	Qualitative forecasting function
f^{qn}	Quantitative forecasting function
F_{cal}	Calendar forecasting function
\mathbf{F}	Regressor
\mathbf{F}^{ql}	Piece-wise qualitative forecasting function
\mathbf{F}^{qn}	Piece-wise quantitative forecasting function

h	Short-term prediction horizon length in unitary basis
H	Short-term prediction horizon periods of size τ
I	Irregular fluctuations component
k	Number of clusters
κ	Number of neighbours
K	Set of cluster labels
K_{test}	Set of test cluster labels
ℓ	Level component of Holt-Winters
L	Lag operator
LB	Ljung-Box coefficient
LS	Labeled sample of RBFMMP+OR
max	Maximum value function
M	Number of hidden neurons of an ANN
Mo	Statistical Mode
m	Delay vector size
n	Number of vector elements
N	Number of qualitative or quantitative vector elements
N'	Number of S samples
η	Random noise with null autocorrelation
O	Indexes set
p, d, q, P, D, Q	SARIMA polonomial coefficients
p_s	Number of samples of period τ
P	Set of prototypes
\mathcal{P}	Partition
\mathbb{P}	Prediction of the qualitative components
Pr	Probability
Q_k	Separability coefficient
R	Random component
R_T	Residual vector at time T
s^1, s^2	Seasonal components for Holt-Winters
s	Index of the last occurrence of any class
\tilde{s}	Seasonality for SARIMA
s_{sig}	Noiseless signal value
\hat{s}_{sig}	Estimated noiseless signal value
$s(i)$	Similarity of object i
S	Seasonal component

S	Sample of the RBFMMP+OR
S'	Transformed sample of the RBFMMP+OR
S	Sequence of samples for the RBFMMP+OR
t	Index time in unitary basis
τ	Period
τ'	Subperiod
T	Trend component
T	Index time period of τ size basis
TS	Training sample for the RBFMMP+OR
U	Neighbourhood
v	Index of neighbour residual vector
V	Set of residual indexes associated with any mode or class
W'	Queue
\mathbf{x}, \mathbf{y}	Vectors
X_T^a	Qualitative time series element of section a of a period
$\chi^2_{(m)}$	Chi-squared distribution with m degrees of freedom
\mathbf{X}	Time series of qualitative or unitary patterns
\hat{X}	Estimation of unitary pattern
\mathbf{X}^a	Qualitative time series of section a of a period
Y_t	Time series element
Y'_t	Differenced time series value
\hat{Y}_{t+1}	Prediction at next instant
\mathbf{Y}	Time series vector
\mathbf{Y}_t	Time series vector at time t
\mathbf{Y}'_t	Vector sequence of past observations
$\hat{\mathbf{Y}}_{t+1}$	Time series prediction vector
\mathbf{Y}_{full}	Full time series
Z'_T	Time series interval T of size τ
\bar{Z}'_T	Mean of interval T of size τ
Z^a_T	Sum of values of section a of interval T
\bar{Z}'	Median of interval of size τ
\mathbb{Z}	Prediction of the quantitative components at any horizon H
Z	Sum of values of the interval Z'
\hat{Z}	Sum of values estimation of the next interval τ

\mathbf{Z}	Quantitative time series
\mathbf{Z}^a	Quantitative time series of section a of a period
ρ_τ	Coefficient of the simple autocorrelation at lag τ
$\hat{\pi}_{t,t}$	Coefficient of partial autocorrelation at lag t
$\alpha, \beta, \mathbf{w}, w$	Weights
$\acute{\alpha}$	Limit of past observations
Φ	AR coefficient
ϕ	AR seasonal coefficient
Θ	MA coefficient
θ	MA seasonal coefficient
ϵ	Estimation error
ε	Neighborhood size
μ	Mean, cluster centroid
Δn	Time increment
σ^2	RBF width in terms of variance
ψ	Radial basis function
Ψ	Mapping function

Bibliography

- [AF14] S. Alvisi and M. Franchini. Assessment of the predictive uncertainty within the framework of water demand forecasting by using the model conditional processor. In *Procedia Engineering*, volume 89, pages 893–900, 2014. 14
- [AFM07] C. Alvisi, M. Franchini, and A. Marinelli. A short-term, pattern-based model for water-demand forecasting. *Journal of Hydroinformatics*, 9(1):39–50, 2007. 11, 59
- [AH01] S. Aksoy and R. M. Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, 22(5):563–582, 2001. 56
- [AM10] R. Arce and R. Mahia. Modelos arima. *U.D.I. Econometria e Informatica*, page 31, 2010. 36
- [AS77] E. Alperovits and U. Shamir. Design of optimal water distribution systems. *Water Resources Research*, 13(6):885–900, 1977. 6
- [AYK⁺11] D. Abhijit, S. V. Yan, R. Inke König, J. E. Bailey-Wilson, and J. D. Malley. Brief review of regression-based and machine learning methods in genetic epidemiology: The Genetic Analysis Workshop 17 experience. *Genetic Epidemiology*, 35, 2011. 65
- [Bat00] R. Batchelor. The imf and oecdversus consensus forecasts. *Applied Economics*, 33:225–235, 2000. 14
- [BC13] K. Benmouiza and A. Cheknane. Forecasting hourly global solar radiation using hybrid k-means and nonlinear autoregressive neural network models. *Energy Conversion and Management*, 75:561–569, 2013. 12, 59

- [BJR94] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 3rd edition, 1994. 8, 29, 35
- [Bla08] P. E. Black. Finite State Machine. *Dictionary of Algorithms and Data Structures*, 2007:1–7, 2008. 21
- [BM91] R. S. Boyer and J. S. Moore. MJRTYA Fast Majority Vote Algorithm. In *Automated Reasoning*, pages 105–117. Springer, 1991. 48
- [BP70] G. E. P. Box and D. A. Pierce. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970. 35
- [BP86] J. S. Bendat and A. G. Piersol. *Random Data: Analysis and Measurement Procedures*. John Wiley and Sons, Inc, 1986. 33
- [BRRP11] D. Barcelli, C. Rocchi, G. Ripaccioli, and C. A. Pascucci. Decentralized wireless control of large-scale systems. *Journal of Process Control*, 2011. 6
- [BVVS13] M. Bakker, J. Vreeburg, and K. Van Schagen. A fully adaptive forecasting model for short-term drinking water demand. *Environmental Modelling & Software*, 48:141–151, 2013. 13, 106
- [BWWA01] M. Blix, J. Wadefjord, U. Wiencke, and M. Adahl. How good is the forecasting performance of major institutions?“. *Economic Review of the Swedish Central Bank*, 2001. 14
- [BWZ08] M. Brio, G. M. Webb, and A. R. Zakharian. Discretization methods. *Mathematics in Science and Engineering*, 213(C):59–108, 2008. 58
- [CCK⁺08] P. Cutore, a. Campisano, Z. Kapelan, C. Modica, and D. Savic. Probabilistic prediction of urban water consumption using the SCEM-UA algorithm. *Urban Water Journal*, 5(2):125–132, 2008. 14
- [CGB96] M. R. Chmielewski and J. W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15(4):319–331, 1996. 58
- [Cle93] W. S. Cleveland. *Visualizing Data Why do Data Visualization ?*, volume 36. Hobart Press, Summit, New Jersey, 1993. 29
- [CLX⁺11] D. S. Cao, Y. Z. Liang, Q. S. Xu, Q. N. Hu, L. X. Zhang, and G. H. Fu. Exploring nonlinear relationships in chemical data using kernel-based methods. *Chemometrics and Intelligent Laboratory Systems*, 107(1):106–115, 2011. 58

- [CM00] E. Castano and L. F. Melo, Melo-Velandia. Métodos de combinación de pronósticos: Una aplicación a la inflación colombiana. *Lecturas de Economía*, 2000(52):113–165, 2000. 8, 9
- [Dat16] Consensus economics: Worlds leading macroeconomics survey firm. <http://www.consensuseconomics.com/>, 2016. [On-line; accessed February 2016]. 14
- [Des13] B. Desgraupes. *Clustering Indices*. University Paris Ouest, Lab ModalX, 2013. [Accessed July 2015]. 54
- [DI07] R. H. Dieck and Systems Instrumentation. *Measurement uncertainty: methods and applications*. ISA Research Triangle Park, NC, 2007. 80
- [DMSR14] Emmanuel a Donkor, Thomas a Mazzuchi, Refik Soyer, and J Alan Roberson. Urban Water Demand Forecasting : Review of Methods and Models. *Journal of Water Resources Planning and Management*, 140(February):146–159, 2014. 14
- [ELW⁺08] K. Eneman, H. Luts, J. Wouters, M. Bachler, N. Dillier, W. Dreschler, M. Froehlich, G. Grimm, V. Hohmann, R. Houben, A. Leijon, A. Lombard, D. Mauler, M. Moonen, H. Puder, M. Schulte, A. Spriet, and M. Vormann. Evaluation of signal enhancement algorithms for hearing instruments. In *European Signal Processing Conference*, 2008. 56
- [FKCB84] A. Fink, J. Kosecoff, M. Chassin, and R. H. Brook. Consensus methods: Characteristics and guidelines for use. *American Journal of Public Health*, 74(9):979–983, 1984. 14
- [Fra85] J. Franke. A levinson-durbin recursion for autoregressive-moving average processes. *Biometrika*, 72(3):573–581, 1985. 35
- [Fre95] Y. Freund. Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, 121(2):256–285, 1995. 48
- [GE06] I. Guyon and A. Elisseeff. An Introduction to Feature Extraction. In Isabelle Guyon, Masoud Nikravesh, Steve Gunn, and Lotfi A Zadeh, editors, *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 1–25. Springer Berlin Heidelberg, 2006. 62, 64
- [Gib83] J.D. Gibbons. Nonparametric methods for quantitative analysis. *Technometrics*, 28(2):275–277, 1983. 33
- [Goo10] P. Goodwin. The Holt-Winters Approach to Exponential Smoothing : 50 Years Old and Going Strong. *Foresight*, pages 30–33, 2010. 40

- [GS10] C. M. Grinstead and J. L. Snell. Markov Chains. *Introduction to Probability*, pages 1–66, 2010. 22
- [HA13] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. Otexts, New York, 1st edition, 2013. 3, 26, 37
- [HH92] J.C. Hoskins and D.M. Himmelblau. Process control via artificial neural networks and reinforcement learning. *Computers and Chemical Engineering*, 16(4):241 – 251, 1992. Neural network applications in chemical engineering. 49
- [HK15] C. J. Hutton and Z. Kapelan. A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting. *Environmental Modelling and Software*, 66:87–97, 2015. 14
- [HKP11] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. 105
- [HM88] E. J. Hannan and A. J. McDougall. Regression procedures for arma estimation. *Journal of the American Statistical Association*, 83(402):pp. 490–498, 1988. 39
- [HPS01] S. H. Hippert, C.E. Pedreira, and R. Castro Souza. Neural networks for short-term load forecasting: A review and evaluation. *Power Systems, IEEE Transactions on*, 16(1):44–55, 2001. 6
- [Jon14] R. C. Jones. Making better (investment) decisions. *Institutional Investor Journals*, pages 128–143, 2014. 14
- [JQWS15] Ping Jiang, Shanshan Qin, Jie Wu, and Beibei Sun. Time Series Analysis and Forecasting for Wind Speeds Using Support Vector Regression Coupled with Artificial Intelligent Algorithms. *Mathematical Problems in Engineering*, 2015:1–14, 2015. 3
- [JV08] A.S. Johannes and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press, 2008. 64
- [Kal04] P. S. Kalekar. Time series forecasting using Holt-Winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, pages 1–13, 2004. 40, 43, 97
- [KAS⁺03] T. Kempowsky, M. Aquilar, J. Subias, Le Lann, and M.V. Classification tool based on interactivity between expertise and self-learning techniques. In *IFAC Safeprocess*, 2003. 11

- [KBB05] P. Krause, D. P. Boyle, and F. Bäse. Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, 5:89–97, 2005. 19
- [Kem04] T. Kempowsky. *SALSA Situation Assessment using LAMDA Classification Algorithm*. LAAS-CNRS, 2004. 11
- [KOB13] Devin C Koestler, Hernando Ombao, and Jesse Bender. Ensemble-based methods for forecasting census in hospital units. *BMC medical research methodology*, 13:67, 2013. 14
- [KP10] M. Kumar and Nitin R. Patel. Using clustering to improve sales forecasts in retail merchandising. *Annals OR*, 174(1):33–46, 2010. 59
- [KS04] H. Kantz and T. Schreiber. Nonlinear Time Series Analysis. *Technometrics*, 47(3):369, 2004. 48, 57
- [LB78] G. M. Ljung and G. E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978. 35
- [LP08] M. Leutbecher and T.N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. 13
- [LPR15] R. Lopez, V. Puig, and H. Rodriguez. An implementation of a multi-model predictor based on the qualitative and quantitative decomposition of the time-series. In *ITISE 2015 (International work-conference on Time Series)*, volume 1 of 2, Granada, Spain, 2015. 74
- [LRS13] G. Lindgren, H. Rootzen, and M. Sandsten. *Stationary stochastic processes for scientists and engineers*. Chapman and Hall, 2013. 32
- [MALS⁺07] F. Martinez-Alvarez, A. Troncoso Lora, J. Santos, Cristobal, Riquelme, and J. Santos, Riquelme. Partitioning-clustering techniques applied to the electricity price time series. In Hujun Yin, Peter Tio, Emilio Corchado, William Byrne, and Xin Yao, editors, *IDEAL*, volume 4881 of *Lecture Notes in Computer Science*, pages 990–999. Springer, 2007. 60
- [McN87] S. K. McNees. Consensus forecasts: tyranny of the majority? *New England Economic Review*, 50(Nov):15–21, 1987. 13
- [MG08] Q. Min and Peter Z. G. Trend time-series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*, 19:808–816, 2008. 65
- [MJK08] D. C. Montgomery, Cheryl L. Jennings, and M. Kulahci. *Introduction to time series analysis and forecasting*. Wiley-Interscience, 1 edition, March 2008. 23

- [Nip11] T. Nipkow. Majority Vote Algorithm Revisited Again. *International Journal of Software and Informatics*, 5:21–28, 2011. 48
- [NR11] F. Novotny and M. Rakova. Assessment of consensus forecasts accuracy: The Czech National Bank perspective. *Finance a Uver - Czech Journal of Economics and Finance*, 61(4):348–366, 2011. 14
- [OJ06] V. Overloop and P. Jules. *Model predictive control on open water systems*. IOS Press, 2006. 6
- [OL94] L. E. Ormsbee and K. E. Lansey. Optimal control of water supply pumping systems. *Journal of Water Resources Planning and Management*, 120(2):237–252, 1994. 6
- [Org93] World Health Organization. *Guidelines for drinking-water quality: recommendations*, volume 1. World Health Organization, 1993. 5
- [Orr96] M. J. L. Orr. Introduction to radial basis function networks. *University of Edinburg*, pages 1–7, 1996. 49
- [PRP⁺13] J. Pascual, J. Romera, V. Puig, G. Cembrano, R. Creus, and M. Minoves. Operational predictive optimal control of Barcelona water transport network. *Control Engineering Practice*, 21(8):1020–1034, 2013. 6
- [QPG⁺10] J. Quevedo, V. Puig, Cembrano G., J. Blanch, J. and Aguilar, D. Saporta, G. Benito, M. Hedo, and A. Molina. Validation and reconstruction of flow meter data in the barcelona water distribution network. *Control Engineering Practice*, 18(6):640 – 651, 2010. 59, 60, 96
- [QSPB14] J. Quevedo, J. Saludes, V. Puig, and J. Blanch. Short-term demand forecasting for real-time operational control of the barcelona water transport network. In *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, pages 990–995, June 2014. 11, 26, 74
- [Ras14] S. Raschka. *Python Machine Learning*. Packt Publishing, 2014. 56
- [Rou87] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. 55
- [S⁺98] D. Stephenson et al. *Water supply management*. Kluwer Academic Publishers, 1998. 6

- [SBM07] R. C. Souza, M. Barros, and C. V. C. Miranda. Short term load forecasting using double seasonal exponential smoothing and interventions to account for holidays and temperature effects. *TIAIO II-2 do Taller Latino Iberoamericano de Investigación de Operaciones, Acapulco, México*, 2007. 45
- [Sch13] S. Schneider. *Simple Linear Regression*. SAGE Publications Ltd, 2013. 29
- [SCM02] C. D. Stefano, A. D. Cioppa, and A. Marcelli. An adaptive weighted majority vote rule for combining multiple classifiers. *Object recognition supported by user interaction for service robots*, 2(FEBRUARY):192–195, 2002. 48
- [Ser15] National Weather Service. Ensemble prediction systems). <http://www.wpc.ncep.noaa.gov>, 2006 (Last access: Sept 6, 2015). 14
- [She05] R. Shepard. A general nonlinear expansion form for electronic wave functions. *Journal of Physical Chemistry A*, 109(50):11629–11641, 2005. 58
- [SJ97] M. R. Smith and T. A. (Eds.) Johansen. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, London, 1997. 9, 49, 60
- [SKF08] A. Stolcke, S. Kajarekar, and L. Ferrer. Nonparametric feature normalization for SVM-based speaker verification. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1577–1580, 2008. 56
- [SS11] R.H. Shumway and D.S. Stoffer. Arima models. In *Time Series Analysis and Its Applications*, Springer Texts in Statistics, pages 83–171. Springer New York, 2011. 45
- [SSF93] L. Shvartser, U. Shamir, and M. Feldman. Forecasting Hourly Water Demands by Pattern Recognition Approach, 1993. 10
- [Str94] S. H. Strogatz. *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity)*. Studies in nonlinearity. Perseus Books Group, 1994. 20
- [TA15] M. K. Tiwari and J. F. Adamowski. An ensemble wavelet bootstrap machine learning approach to water demand forecasting: a case study in the city of calgary, canada. *Urban Water Journal*, pages 1–17, 2015. 14
- [Tc11] F. Tak-chung. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164 – 181, 2011. 9

- [TdMM06] J. W. Taylor, L. M. de Menezes, and P. E. McSharpy. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22(1):1–16, 2006. 44
- [Tho80] N. T. Thomopoulos. *Applied Forecasting Methods*. Prentice-Hall, 1980. 42
- [U.S14] U.S. Energy Information Administration. March 2015 Monthly Energy Review. *U.S. Energy Information Administration*, pages 151–152, 2014. 3
- [War05] T. Warren-Liao. Clustering of time series dataa survey. *Pattern Recognition*, 38(11):1857–1874, 2005. 57
- [Wat03] Aquastat. <http://goo.gl/6D3umv>, July 2003. [Online; Accessed February 2016]. 3
- [XKQ⁺08] K. Xindong, V. Kumar, J. Quinlan, Ross., G. Joydeep, Yang Q., H. Motoda, G. J. McLachlan, A. Ng, L. Bing, P. S. Yu, Z. H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining, 2008. 64
- [XPX11] W. Xiaochen, G. Peng, and H. Xiaobin. A review of wind power forecasting models. *Energy Procedia*, 12:770 – 778, 2011. The Proceedings of International Conference on Smart Grid and Clean Energy Technologies (ICSGCE 2011). 3
- [YBH⁺12] Ma. Yudong, F. Borrelli, Brandon Hency, Brian Coffey, Sorin Bengea, and Philip Haves. Model predictive control for the operation of building cooling systems. *Control Systems Technology, IEEE Transactions on*, 20(3):796–803, 2012. 6
- [ZAWL02] S. L. Zhou, McMahon T. A., A. Walton, and J. Lewis. Forecasting operational demand for an urban water supply zone. *Journal of Hydrology*, 259(14):189 – 202, 2002. 13



Unless otherwise expressly stated, all original material of whatever nature created by Rodrigo López Farías and included in this thesis, is licensed under a Creative Commons Attribution Noncommercial Share Alike 2.5 Italy License.

Check creativecommons.org/licenses/by-nc-sa/2.5/it/ for the legal code of the full license.

Ask the author about other uses.